Learning Safe Control for Multi-Robot Systems: Methods, Verification, and Open Challenges

Kunal Garg*, Songyuan Zhang, Oswin So, Charles Dawson, Chuchu Fan

[a]*Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, 02139, MA, USA*

Abstract

In this survey, we review the recent advances in control design methods for robotic multi-agent systems (MAS), focusing on learning-based methods with safety considerations. We start by reviewing various notions of safety and liveness properties, and modeling frameworks used for problem formulation of MAS. Then we provide a comprehensive review of learning-based methods for safe control design for multi-robot systems. We start with various shielding-based methods, such as safety certificates, predictive filters, and reachability tools. Then, we review the current state of control barrier certificate learning in both a centralized and distributed manner, followed by a comprehensive review of multi-agent reinforcement learning with a particular focus on safety. Next, we discuss the state-of-the-art verification tools for the correctness of learning-based methods. Based on the capabilities and the limitations of the state-of-the-art methods in learning and verification for MAS, we identify various broad themes for open challenges: how to design methods that can achieve good performance along with safety guarantees; how to decompose single-agent-based centralized methods for MAS; how to account for communication-related practical issues; and how to assess transfer of theoretical guarantees to practice.

Keywords: Safe Multi-agent Reinforcement Learning; Certificate-based Multi-Agent Control; Verification for Multi-Agent Systems

*Corresponding author

Email address: kgarg@mit.edu (Kunal Garg)

Contents

1. Introduction

1.1. Motivation and applications of MAS

Multi-agent systems (MAS) have received tremendous attention from scholars in different disciplines, including computer science and robotics, as a means to solve complex problems by subdividing them into smaller tasks [1]. Some examples of MAS include smart grids [2], search and rescue teams [3, 4], edge computing [5], wireless communication networks [6], space systems [7, 8, 9], package delivery [10], power systems [11], and micro-grids [12]. The design and analysis of MAS controllers present unique challenges, such as scalability, verification, and robustness to factors such as communication issues, adversarial or non-cooperative agents, and partial observability. While we will provide a detailed discussion on the limitations and challenges of learning-based methods for MAS, interested readers on current broad challenges in various aspects of MAS are referred to [13, 14, 15, 16]. Learning-based methods have been successfully deployed on multi-robot systems demonstrating collision-free safe behaviors in physical robots, e.g., for drones in [17] and ground vehicles in [18, 19]. Some recent works have shown the benefits of learning-based methods compared to *classical* methods for robotic MAS-specific tasks such as cooperative exploration [20], where learning-based methods can achieve coverage of an unknown region in half the time on a hardware robot car platform. Inspired by these recent successes of learning-based methods in safe MAS control, we provide a comprehensive summary of the current state-of-the-art learning-based methods for safe MAS control.

1.2. Scope of this survey

The four major topics of focus in this survey are

Shielding-based methods — Section 3: Methods that delegate safety to shielding function or safety filter which preserves the safety of the MAS. These include non-learning-based Control Barrier Function (CBF), Predictive Safety Filters (PSF), Hamilton-Jacobi (HJ) Reachability, Automata-based methods, and some heuristic methods.

Methods for learning CBF — Section 4: Methods for learning centralized and distributed CBF along with a control policy for MAS.

Multi-agent reinforcement learning (MARL) — Section 5: Methods that apply reinforcement learning to MAS and directly tackle safety constraints.

Verification of learning-enabled MAS — Section 6: The challenges inherent in verifying learned controllers for MAS, how various centralized verification tools have been extended to handle MAS, and communication issues specific to MAS.

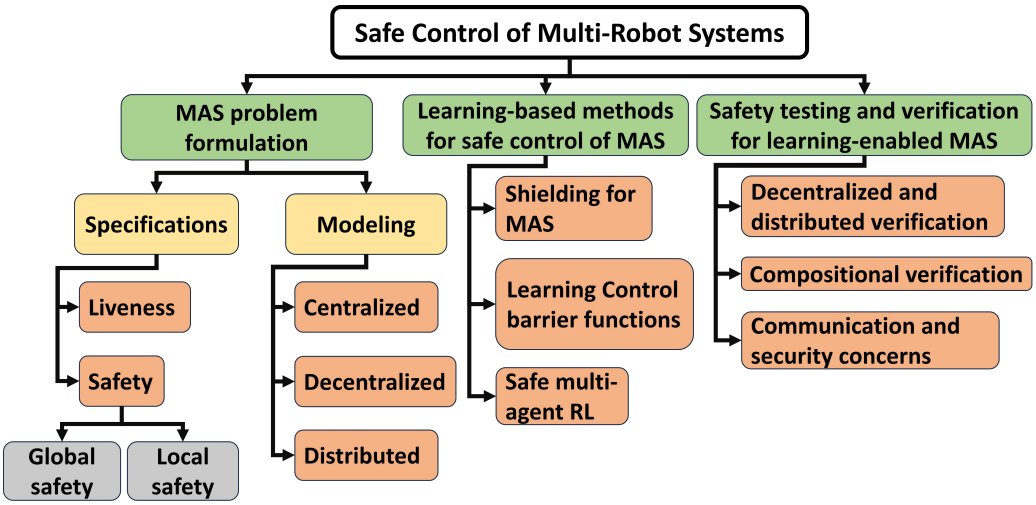

Figure 1: Overview of the survey taxonomy on safe learning for MAS.

1.3. Main takeaway from this survey

- *Focus on MAS safety*: This is the first survey on robot MAS with an explicit focus on safety. The survey walks the reader through the taxonomy of the MAS control design problems, various learning-based methods for safe control synthesis, and the open problems in the field of safe MAS control.

- *Comprehensive survey of learning-based methods for multi-robot systems*: Unlike numerous existing surveys that only focus on MARL, this survey is intended to be a starting point for researchers getting started with learning-based safe control of MAS with a discussion of the capabilities and limitations of the existing learning-based tools.

- *Vision for the future of safe MAS control*: The survey identifies a range of open problems in the field of safe learning-based control, and verification thereof, for robot MAS and it informs the future research on the topic.

1.4. Previous surveys on multi-agent systems

Many survey and review articles appeared in the past few years on the topic of MAS. However, the key elements that differentiate this survey from the prior work are: 1) its focus on the safety of robotic MAS, 2) general learning methods as the central theme, and 3) its identification of open problems and challenges in safe learning-based MAS research. The article [21] provides a detailed introduction to MAS taxonomy and control and [22] on Unmanned Aerial Vehicle swarms, but the discussion in these articles is restricted to non-learning-based methods. Given the popularity of MAS and safety in the context of RL, there have been many surveys that cover RL for MAS [23, 24, 25, 26, 27, 28, 29] or for single-agent safety [30]. Of these, only a few surveys [27, 31] cover the intersection of both topics. Despite this, many MARL works acknowledge safe MARL as a new area that has not been explored much but is a promising future direction [24, 26, 31].

The topic of safety has been reviewed extensively in the robotics community [32]. However, these works focus on the single-agent case, ignoring a broad category of disturbances and safety issues that are particular to MAS (e.g., communication delay and errors). While [30] discusses safety but for single-agent systems. Quite a few surveys are looking at distributed optimization [11, 12, 33], power systems [11] and micro-grids [12]. While safety is not explicitly discussed in these works, some of the methods reviewed in these surveys can incorporate it via the inclusion of safety constraints. Finally, many surveys are focusing on applications of MAS [4, 34, 33, 12].

1.5. Topics not covered by this article

Given the main focus of the survey being safe learning-based methods for MAS, various topics about robotic MAS are out of the scope of this paper. A non-comprehensive list of such topics along with recent surveys on those topics is Vector field-based methods [35, 36]; Model predictive control [37]; Consensus control [38]; Distributed optimization; [33, 39]; and Multi-agent games [40].

1.6. Organization

We start with a general problem formulation, notations, and common definitions for MAS in Section 2. Section 3 introduces shielding methods for the safety of MAS, then Section 4 discusses learned certificates more specifically. Section 5 discusses various MARL-based methods. Section 6 covers verification techniques for MAS. Sections 7 and 8 conclude with a discussion of the challenges and open problems in the field of safe MAS control.

2. MAS problem formulation

This section defines common notions used in the context of multi-agent systems (MAS); namely, agent models, specifications, and modeling frameworks.

2.1. Definitions and notations

In this work, we focus on a general class of MAS consisting of N agents where each agent is a dynamic system modeled as

$$\dot{x}_i = F(x_i, u_i, d_i, \nu_{ij}(x_j)), \quad x_i \in \mathcal{X}_i, \quad u_i \in \mathcal{U}_i, \tag{1}$$

where $x_i \in \mathbb{R}^{n_i}, u_i \in \mathbb{R}^{m_i}$ denote the state and the input of agent i. The sets $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}, \mathcal{U}_i \subseteq \mathbb{R}^{m_i}$ denote the operational workspace and the set of inputs for the i-th agent. The term $d_i \in \mathcal{D}_i$ denotes the disturbances, uncertainties, and unmodeled dynamics for agent i, while the map $\nu_{ij} : \mathbb{R}^{n_j} \to \mathbb{R}^{p_i}$ denotes the influence of the other agents $j \neq i$ or in other words, the inter-agent coupling of the MAS dynamics. The joint state vector and the input vector are denoted as $\mathbf{x} = [x_1^\top, x_2^\top, \ldots, x_N^\top]^\top \in \mathcal{X} \subset \mathbb{R}^{\sum n_i}$ and $\mathbf{u} = [u_1^\top, u_2^\top, \ldots, u_N^\top]^\top \in \mathcal{U} \subset \mathbb{R}^{\sum m_i}$, respectively. The combined dynamics of the MAS can be written as

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u}, \mathbf{d}). \tag{2}$$

To enable theoretical analysis, it is typically assumed that the function $\mathbf{F}$ is locally Lipschitz continuous [41].

For robotic systems, let $x_i \supset p_i \in \mathbb{R}^3$ denote the physical location of the i-th agent in the 3D space. The state trajectory of agent i under a control policy π_i starting at an initial condition $x_i(0) \in \mathcal{X}_i$ is denoted as $\phi_i(\cdot, \pi_i; x_i(0)) : \mathbb{R}_+ \to \mathbb{R}^{n_i}$. Correspondingly, the state trajectory of the MAS is denoted as $\Phi(\cdot, \boldsymbol{\pi}; \mathbf{x}(0))$ with $\boldsymbol{\pi} : \mathcal{X} \to \mathcal{U}$ being the joint policy for the

MAS. When an explicit emphasis on the underlying policy is not required, we denote the trajectories of the agents with $x_i(\cdot)$ and that of the MAS with $\mathbf{x}(\cdot)$ for the sake of brevity.

A network can be defined for the MAS with agents denoting the nodes and their communication links denoting edges. Let $R_i > 0$ be the sensing/communication radius of the i-th agent and define $\mathcal{N}_i(t) = \{j \mid \|p_i - p_j(t)\| \leq R\}$ as the set of *neighbors* of the i-th agent, i.e., the set of agents from (respectively, to) which the agent i can receive (respectively, send) information [42]. Using this notion of neighbors, a graph topology can be defined for the MAS as $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ where $\mathcal{V} = \{1, 2, \ldots, N\}$ is the set of vertices denoting the agents and $\mathcal{E}(t)$ the time-varying set of connections given as $\mathcal{E}(t) = \{(i, j) \ : \ j \in \mathcal{N}_i, i \in \mathcal{V}\}$, that is, there is an edge $\mathcal{E}_{ij}(t)$ from agent j to agent i at time t if the agent i is able to receive information from agent j. A time-varying adjacency matrix $\mathcal{A}$ for this graph is defined as

$$\mathcal{A}_{ij}(t) = \begin{cases} 1 & j \in \mathcal{N}_i(t), \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The Laplacian matrix $\mathcal{L}$ corresponding to the adjacency matrix $\mathcal{A}(t)$ is given as

$$\mathcal{L}_{ij}(\mathcal{A}(t)) = \begin{cases} \sum\limits_{k \neq i} \mathcal{A}_{ik}(t), & j = i, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

From [43, Theorem 2.8], we know that the graph topology $\mathcal{G}(t)$ is connected at time t if and only if the second smallest eigenvalue of the Laplacian matrix is positive, i.e., $\lambda_2(\mathcal{L}(\mathcal{A}(t))) > 0$.

2.2. MAS specifications

In the temporal logic language [44, Chapter 3], the control objective for the MAS can be characterized as:

1. **Safety property**: *Something bad never happens.* Agents remain in the safe region $\mathcal{S}(t) \subset \mathbb{R}^{\sum n_i}$ at all times, i.e., $\mathbf{x}(t) \in \mathcal{S}(t)$ for all $t \geq 0$ [45, 46]. Generally, the safe region is defined as the complement of the occupancy set of other agents, obstacles, and restricted regions in the workspace.

2. **Liveness property**: *Something good will eventually happen.* Agents move towards minimizing[1] a (possibly joint) objective function $\Psi : \mathbb{R}^{\sum n_i} \to \mathbb{R}$, i.e., $\mathbf{x}(t) \to \text{argmin}_{\mathbf{x}} \Psi(\mathbf{x})$ [47, 48, 49, 50, 51]. The most common example of a liveness property is each agent required to reach a goal location.

Here, we use the term dynamic obstacles for agents or entities that do not follow the designed control policy. Some safety properties can be decomposed at the agent level, while some safety properties need to be stated for the MAS as a whole. For example, the inter-agent safety property needs to be specified for the MAS using a *global* safe set $\mathcal{S}$ while obstacle avoidance property can be expressed individually for each agent with a *local* safety set $\mathcal{S}_i$. Another important requirement that can be posed as a safety property in MAS is *connectivity* maintenance, which becomes an important property for various applications such as coverage [52] and formation control [53]. For the sake of performance criteria as well as maintaining safety, the agents in MAS need to sense each other and might also need to actively communicate certain information. A detailed discussion on various safety properties is provided next.

Note that there are various notions of safety used in the literature, however, since the focus of this survey is robotic MAS, we restrict our discussions to safety as it pertains to *physical* safety of the agents.

2.2.1. Safety property

For a MAS (1), the notion of safety is defined as follows.

Definition 1 (**Safety**)**.** *Given a (potentially time-varying) safety constraint set $\mathcal{S}(t)$, the MAS (1) is safe with respect to $\mathcal{S}(t)$ if for all $\mathbf{x}(0) \in \mathcal{S}(0)$, the trajectories satisfy $\mathbf{x}(t) \in \mathcal{S}(t)$ for all $t \geq 0$.*

Below, we give examples of some of the possible safety constraints for robotic MAS. We draw a distinction between properties that require considering the joint state of the MAS (global properties) and those that can be checked using only individual agent states (local properties).

1. Global MAS safety properties

[1] In some works, the objective is given as maximization instead of minimization.

- *Inter-agent collision avoidance*: Given a safe distance $0 < r_s < R$, the inter-agent collision avoidance can be formulated through $\mathcal{S} = \{\mathbf{x} \mid \|p_i - p_j\| > r_s, j \neq i\}$ [54, 55].
- *Connectivity maintenance*: Given a communication radius $R > 0$, the MAS network connectivity maintenance can be formulated as $\mathcal{S} = \{\mathbf{x} \mid \lambda_2(\mathcal{L}(\mathcal{A}(\mathbf{x}))) > 0\}$ [42, 56], where

$$\mathcal{A}_{ij}(\mathbf{x}) = \begin{cases} 1, & \|p_i - p_j\| \leq R, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

2. Local safety properties

- *Obstacle avoidance*: Given a safe distance $0 < r_s < R$ and a set of (potentially moving) obstacles $\mathcal{O}_j(t) \subset \mathbb{R}^3$, obstacle avoidance can be formulated through the set $\mathcal{S}_i(t) = \{p_i \mid \text{argmin}_{p \in \mathcal{O}_j(t)} \|p_i - p\| > r_s, \forall j\}$ [55, 57].
- *State limits*: Given state limits (e.g., positions and velocities) of the form $x_m^j \leq x_i^j \leq x_M^j$ where x_i^j denotes the $j-$th component of x_i, the safety can be formulated with the set $\mathcal{S}_i = \{x_i \mid x_m^j \leq x_i^j \leq x_M^j, j \in \{1, 2, \ldots, n_i\}\}$[2] [58, 59].

2.2.2. Liveness properties

Here, we discuss commonly studied liveness properties for a MAS. [3]:

1. *Consensus*: Given a consensus point x_c, the trajectories of each of the agents converge in the following sense: $\lim_{t \to \infty} x_i(t) = x_c$ [60, 61].[4]
2. *Formation*: Given a set of off-set vectors x_{ij} for each (i, j), $i \neq j$, the trajectories of the MAS satisfy $x_i(t) - x_j(t) \to x_{ij}$ as $t \to \infty$ [62, 63].
3. *Coverage*: Given an exploration region $\mathcal{X}_e \subset \mathcal{X}$ and a distribution function $\varphi : \mathcal{X}_e \to \mathbb{R}_+$ and a smooth increasing function $f : \mathbb{R} \to \mathbb{R}$ that

[2] More general constraints of the form $r(x_i) \leq 0$ for some constraint function r can also be considered.

[3] Some of these properties require compatible workspaces for the agents, i.e., $n_i = n_j = n$ for all i.

[4] More generally, it is possible to define consensus to $\rho(x_c)$ for some function ρ.

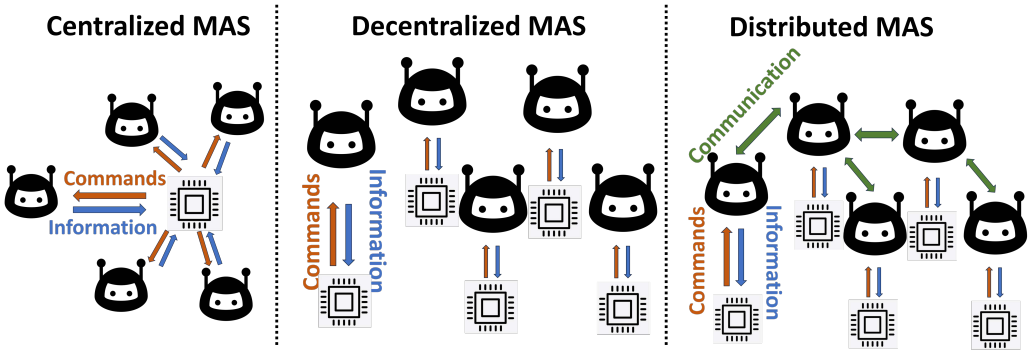

Figure 2: MAS modeling: centralized, decentralized and distributed frameworks.

measure the degradation of sensing performance, the coverage objective is to minimize the function $\Psi(x) = \sum_{i=1}^{N} \int_{V_i} f(\|x_i - z\|)\phi(z)dz$, where $V_i \subset \mathcal{X}_e$ denotes the region where agent i is responsible for coverage [52, 64].

4. *Goal reaching*: Given a set of goal states $x_{gi} \in \mathbb{R}^{n_i}$, the trajectories of each agent reach the goal, i.e., $\lim_{t\to\infty} x_i(t) = x_{gi}$ for each i [65, 54, 66].

5. *Reference tracking*: Given a reference trajectory $x_{i,ref}(\cdot)$, the trajectories of each of the agent satisfy $x_i(t) - x_{i,ref}(t) \to 0$ as $t \to \infty$ [67, 68].

The liveness properties are important for capturing both performance criteria (e.g. goal reaching or trajectory tracking) and certain safety properties (particularly global safety properties). For example, MAS might be required to reach a consensus or maintain a formation to maintain a connectivity property. While the focus of this survey is on safety properties, we provide brief comments on the ability of the reviewed methods to accomplish tasks that often require both safety and liveness properties (particularly, goal-reaching).

2.3. MAS modeling framework

2.3.1. Centralized, decentralized, and distributed MAS

In this section, we present various modeling frameworks for MAS. Generally, MAS is modeled under the following three paradigms:

1. *Centralized*: An MAS is termed centralized if there is a *central* node where all the information/sensor data from all the agents is collected and decisions for each of the agents are made.

2. *Decentralized*: An MAS is termed decentralized if each agent makes its own decision based on its local information/sensor data *without* communicating with other agents.

3. *Distributed*: An MAS is termed distributed if each agent makes its own decisions based on its local information/sensor data along with information received by active communication with other agents.

We note there is currently no consensus in the literature on the definition of the decentralized and distributed paradigms. For example, [69] defines decentralized MAS where the agents communicate with their local neighbors. The authors in [70] use a framework where agents communicate only when *needed*, calling this approach decentralized communication. In more recent multi-agent reinforcement learning (MARL) work [49], the authors allow the agents to communicate over a time-varying connectivity graph and call the formulation *fully decentralized*. There are many examples of this interchangeable usage of the terms decentralized and distributed in the literature. For the sake of consistency, in this survey, we will stick with the notions as defined above (see [71, 11]).

Some examples of methods using centralized learning frameworks for safe control of MAS are [46, 72, 73, 74, 75], the works in [76, 77] use a decentralized learning framework, and [78, 79] use a distributed learning framework. Centralized Training Decentralized Execution (CTDE) is a related paradigm, where the joint state and other global information are used to train a decentralized policy for each agent that only has access to local information [55, 29].

[5]HJ and Automata-based methods can be applied directly from safety specifications, while PSF (needs valid control-invariant set) and CBF-based (needs a valid CBF) shielding require domain expertise to use.

[6]CBF [76], PSF [80] and automata-based [77] shielding have distributed versions that maintain their safety guarantees in practice. While HJ-based shielding does have a distributed version by considering pairwise interactions [81], the safety guarantees do not hold for the full MAS.

Method	Safety Guarantees		Requirements		
	Theory	Practice	Domain Expertise	Known Dynamics	Distributed Policy
Shielding	✓	✓	✗/✓[5]	✓	✓/✗[6]
Certificate Learning	✓	✗	✗	✓	✓
Unconstrained MARL	✗	✗	✗	✗	✓
Constrained MARL	✓	✗	✗	✗	✓

Table 1: Overview of different methods of handling safety for MAS. The tick mark denotes available features or requirements, while the cross mark denotes missing features or non-requirements. Blue denotes desirable properties, while red denotes undesirable properties.

2.4. Properties of algorithms for MAS Safety

Finally, we discuss some desirable properties of learning-based algorithms for the safe control of MAS and categorize the main themes reviewed in this paper based on these properties (see Table 1):

Safety Guarantees - Theory: Here, we categorize the methods based on the fact that whether, under some suitable assumptions, it results in a safe policy. For instance, unconstrained MARL that relies on penalty-based mechanisms for encoding safety do not provide theoretical guarantees of safety.

Safety Guarantees - Practice: While theoretical guarantees are important, it is more useful to ask whether the assumptions needed to provide those safety guarantees from theory also hold in practice. For example, while certificate learning methods can provide safety guarantees if the certificate can be perfectly learned by the neural network [82], there is no guarantee that this will be the case. Similarly, while some constrained MARL methods guarantee that the policy at each iterate will be safe [83], this relies on the assumption that the value function

is known exactly and that a trust-region optimization can be solved exactly, neither of which is true in practice.

Requirements - Domain Expertise: An important aspect that dictates the ease of usage and wide applicability of a method is whether domain expertise is needed about the specific dynamics or safety constraints to construct supporting tools and methods needed to apply the method. For instance, hand-crafting a CBF as a shield often requires domain expertise, especially under input constraints.

Requirement - Known Dynamics: Another factor that plays an important role in the generalizability and wide applicability of a method is whether the exact form of the dynamics is needed (e.g., for computing jacobians / querying at arbitrary states) to apply the method, or it is sufficient to have black-box evaluations of the dynamics along a set of trajectories. For instance, hand-crafted shielding methods and certificate learning methods require knowledge of the dynamics and its structure (i.e., control-affine), while RL-based methods are model-free and only require black-box evaluations.

Distributed Policy: Finally, and very importantly for large-scale MAS, one needs to ask whether the policy can be deployed in a distributed manner without a central computation/aggregation node. For instance, HJ-based shielding methods require a centralized framework for safety guarantees.

3. Shielding-based learning for MAS

One popular method of providing safety to learning-based methods is via the use of *shielding* or *safety filter*s, where an *unconstrained* learning method is paired with a *shield* or *safety filter*. Such shields are often constructed without learning with the objective of either modifying the input or the output of the learning method to maintain safety. One benefit of shielding-based methods is that safety can be guaranteed during both training and deployment since the shield is constructed prior to training. However, a drawback of some of these methods is that they require domain expertise for the construction of a valid shield, which can be challenging in the single-agent setting and becomes even more difficult for MAS. Other methods can automatically synthesize shields, but face scalability challenges.

Let $\boldsymbol{\pi} : \mathcal{X} \to \mathcal{U}$ denote the *joint* policy for MAS (2) from a learning-based controller without safety considerations. Shielding-based methods define a *shielding function* $\boldsymbol{\Pi} : \mathcal{X} \times \mathcal{U} \to \mathcal{U}$ that takes the output of $\boldsymbol{\pi}$ and returns a shielded output [46]. The level of safety and the type of safety guarantees that can be obtained depends on how the shielding function $\boldsymbol{\Pi}$ is constructed. We provide an overview of different shielding-based methods used to ensure the safety of learning for MAS in Figure 3.

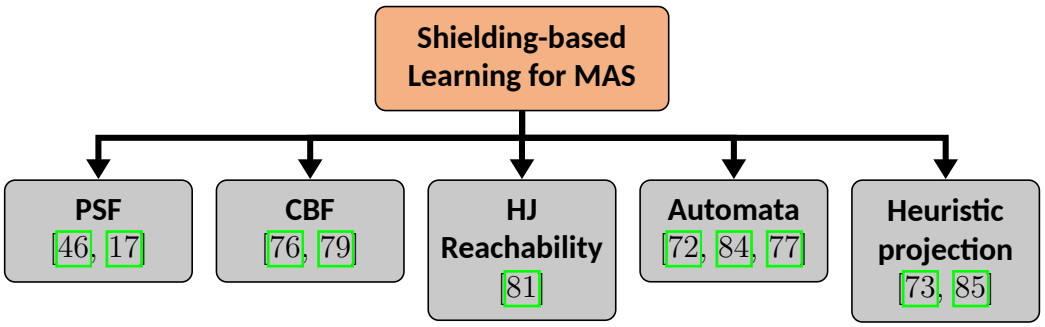

Figure 3: Overview of Shielding-based Learning for MAS.

3.1. Control barrier function-based shielding

One method of constructing a shield $\boldsymbol{\Pi}$ is via a Control Barrier Function (CBF). We start by reviewing the notion of CBF.

3.1.1. Definition of CBF

The notion of CBF was introduced to satisfy the conditions of set invariance, where a set is termed as *forward invariant* if starting in the set, the system trajectories do not leave it [86, 87, 88]. It is also related to the notion of Control Lyapunov Function (CLF) [89, 90] which is commonly used for liveness properties, and extends the definition of barrier certificates [91, 92] to control systems. A comprehensive review of CBFs as a tool for safety can be found in [93].

While there exists various definitions of CBF with slight variations [94, 95, 41, 82], we use the following definition in this survey [93]:

Definition 2 (CBF). *Consider the MAS dynamics (2) with no disturbances, i.e.,* $\mathbf{d} = 0$. *Let* $\mathcal{C} \subset \mathcal{X}$ *be the* 0*-superlevel set of a continuously differentiable*

function $B : \mathcal{X} \to \mathbb{R}$, *i.e.,* $\mathcal{C} = \{\mathbf{x} \in \mathcal{X} : B(x) \geq 0\}$. *Then, the function* B *is a CBF if there exists an extended class-*$\mathcal{K}$ *function* $\alpha : \mathbb{R} \to \mathbb{R}$ [7] *such that:*

$$\sup_{\mathbf{u} \in \mathcal{U}} \left[\frac{\partial B}{\partial \mathbf{x}} \mathbf{F}(\mathbf{x}, \mathbf{u}) + \alpha \left(B(\mathbf{x}) \right) \right] \geq 0, \quad \forall \mathbf{x} \in \mathcal{X}. \tag{6}$$

Safe Control Set: Based on (6), we can define a set of *safe* control input

$$K_{\text{CBF}}(\mathbf{x}) = \left\{ \mathbf{u} \in \mathcal{U} : \frac{\partial B}{\partial \mathbf{x}} \mathbf{F}(\mathbf{x}, \mathbf{u}) + \alpha \left(B(\mathbf{x}) \right) \geq 0 \right\}. \tag{7}$$

The authors in [93] proved the following result on forward invariance of the set $\mathcal{C}$ using the notion of CBF:

Theorem 1. *Let* $\mathcal{C}$ *be the* 0*-superlevel set of a continuous differentiable function* $B : \mathcal{X} \to \mathbb{R}$, *i.e.,* $\mathcal{C} = \{\mathbf{x} \in \mathcal{X} : B(\mathbf{x}) \geq 0\}$. *If* B *is a CBF, and* $\frac{\partial B}{\partial \mathbf{x}} \neq 0$ *for all* $\mathbf{x} \in \partial \mathcal{C}$, *then any Lipschitz continuous policy* $\boldsymbol{\pi} : \mathcal{X} \to \mathcal{U}$ *with* $\boldsymbol{\pi}(\mathbf{x}) \in K_{\text{CBF}}(\mathbf{x})$ *renders the set* $\mathcal{C}$ *forward invariant.*

Remark 1. *The forward invariance of the set* $\mathcal{C}$ *can be used for guaranteeing safety for a set* $\mathcal{S}$ *as follows. Based on Theorem 1, if a CBF and a controller are found on* $\mathcal{X}$ *that satisfy the conditions of Theorem 1 and* $\mathcal{C} \subset \mathcal{S}$, *then starting from any initial condition in* $\mathcal{C}$, *the system remains safe.*

There are variations of CBF that can also render the safety of autonomous systems. For example, an appropriate choice of Lyapunov function can be used for safety since the sublevel sets of a Lyapunov function are forward invariant [96]. Based on this idea, [75] combined CLF and CBF, and introduced a framework for learning a Control Lyapunov Barrier Function (CLBF) that guarantees both safety and stability. The approach in [97] proposes another variation of CBF called the Control Admissibility Model (CAM), which uses a notion of CBF where the function B depends on both the state $\mathbf{x}$ and the control input $\mathbf{u}$. The central idea of most of the variations is still forward invariance as per Theorem 1.

[7] A continuous function $\alpha : \mathbb{R} \to \mathbb{R}$ is said to be an extended class-$\mathcal{K}$ function if it is strictly increasing with $\alpha(0) = 0$.

3.1.2. CBF-based shielding synthesis

For control-affine systems $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}$, condition (6) becomes a linear constraint in the control input $\mathbf{u}$. When the input constraint set $\mathcal{U}$ is a convex polytope, a centralized method to synthesize a safe control input through CBF-based shielding is using Quadratic Programming (QP):

$$\begin{aligned} \min_{\mathbf{u}\in\mathcal{U}} \quad & \|\mathbf{u} - \boldsymbol{\pi}_{\text{nom}}(\mathbf{x})\|^2, \\ \text{subject to} \quad & L_{\mathbf{F}}B(\mathbf{x}) + L_{\mathbf{G}}B(\mathbf{x})\mathbf{u} + \alpha\left(B(\mathbf{x})\right) \geq 0, \end{aligned} \tag{8}$$

where $L_{\mathbf{F}}B(\mathbf{x}) = \nabla B(\mathbf{x})^\top \mathbf{F}(\mathbf{x})$ and $L_{\mathbf{G}}B(\mathbf{x}) = \nabla B(\mathbf{x})^\top \mathbf{G}(\mathbf{x})$ and $\boldsymbol{\pi}_{\text{nom}} : \mathcal{X} \to \mathcal{U}$ is a nominal policy (from some unconstrained learning methods) that does not necessarily consider safety with respect to the set $\mathcal{C}$. Given a state $\mathbf{x}$ and a control policy $\boldsymbol{\pi}_{\text{nom}}$, the CBF-based shield $\boldsymbol{\Pi}$ outputs the solution $\mathbf{u}^*$ to the QP (8) that minimally modifies the nominal control input while guaranteeing the safety of the system. Here, the nominal policy may come from a learning-based approach without safety considerations, and the QP (8) provides a shielding mechanism for enforcing safety with this learned control policy.

3.1.3. Constructing CBF

Once a CBF is given for control-affine systems, we can solve problem (8) for shielding. However, finding a CBF for MAS is not trivial, and there is no generalized framework that can find a CBF for any MAS. Here, we review approaches that can efficiently compute CBFs for certain specific types of systems.

For systems with relatively simple dynamics, such as single integrator, double integrator, and unicycle dynamics, it is possible to use a distance-based CBF [95, 41, 98, 99, 93, 100, 101, 102, 103, 104]. Some works also explore provable safety along with liveness by guaranteeing the feasibility of the underlying CBF-QP [101, 102, 105]. For systems with multiple safety constraints, e.g., velocity constraints, and joint angle constraints, it is possible to design a CBF for each constraint and then combine them [106, 107, 108]. However, one needs domain expertise to handcraft each of these CBFs. It is also difficult to encode input constraints when handcrafting the CBF, and therefore, the CBF-QP (8) can be infeasible.

For systems with polynomial dynamics, it is possible to use the Sum-of-Squares (SoS) method to compute a CBF. The key idea of SoS is that the CBF conditions (6) consist of a set of inequalities, which can be equivalently

expressed as checking whether a polynomial function is SoS. In this manner, a CBF can be computed through convex optimization [109, 110, 111]. However, these methods are limited to polynomial dynamics. Moreover, the SoS-based approaches suffer from the curse of dimensionality (i.e., the computational complexity grows exponentially with respect to the degree of polynomials involved) [109].

3.1.4. Distributed CBF

While centralized CBF is an effective shield for small-scale MAS, due to its poor scalability, it is not easy to use it for large-scale MAS. To address the scalability problem, the notion of distributed CBF can be used [55, 112, 113, 76]. In contrast to centralized CBF where the state $\mathbf{x}$ of the MAS is used, for a distributed CBF, only the local observations and information available from communication with neighbors are used, reducing the problem dimension significantly.

Similar to a variety of notions of centralized CBF, there exists a variety of definitions of distributed CBF in the literature [58, 114, 107, 112, 115, 55]. Following the graph notions of MAS introduced in Section 2.1, we review a slightly modified definition of distributed CBF from [55]. Let $o_i \in \mathcal{O}_i \subset \mathbb{R}^{r_i}$ be the observation vector of agent i, and let $z_i \in \mathcal{Z}_i \subset \mathbb{R}^{q_i}$ be the encoding[8] of the information accepted by agent i from its neighbors $\mathcal{N}_i$. Note that z_i depends on the states of the neighbors of agent i.

Definition 3 (**Distributed CBF**). *Consider the MAS agent dynamics* (1) *with no disturbances and no inter-agent influences, i.e.,* $d_i = 0$ *and* $\nu_{ij}(x_j) = 0$*, for all* $i, j \in \mathcal{V}$*. Let* $\mathcal{C}_i \subset \mathcal{X}_i$ *be the* 0*-superlevel set of a continuously differentiable function* $B_i : \mathcal{X}_i \times \mathcal{O}_i \times \mathcal{Z}_i \to \mathbb{R}$*. Then, the function* B_i *is a distributed CBF if there exists an extended class-*$\mathcal{K}$ *function* $\alpha_i : \mathbb{R} \to \mathbb{R}$ *and for each* $\mathbf{x} \in \mathcal{X}$*, there exists a control input* $\mathbf{u} \in \mathcal{U}$*, such that the following holds:*

$$\frac{\partial B_i}{\partial x_i} F_i(x_i, u_i) + \frac{\partial B_i}{\partial o_i} \dot{o}_i + \sum_{j \in \mathcal{N}_i \cup \{i\}} \frac{\partial B_i}{\partial z_i} \frac{\partial z_i}{\partial x_j} F_j(x_j, u_j) + \alpha_i(B_i(x_i)) \geq 0, \quad \forall i. \tag{9}$$

[8] There are many ways of encoding information, such as concatenation [58], summation [115], and applying attention [55].

Similar to the centralized CBF, under certain conditions, the distributed CBF can also guarantee the safety of the MAS [55]. According to Definition 3, the MAS is assumed to be cooperative, i.e., all the agents coordinate to satisfy CBF condition (9) for MAS [45, 114, 115, 116, 55]. Other works consider the worst-case scenario that the agents are non-cooperative [58], in which case, the agents do not have any communication, resulting in a decentralized CBF formulation. In addition, [117] introduces graph control barrier functions (GCBFs), which not only can certify safety in MAS but also can generalize to an arbitrary number of agents.

3.1.5. Constructing distributed CBF

One way to construct distributed CBF is by *decomposing* centralized CBF. There are many ways to decompose centralized CBF. For example, assuming that other agents keep constant velocities, actively chasing the *ego* agent, or actively avoiding collision with the ego agent [58]. Other works [45, 114, 118, 119] consider risk allocation among agents while decomposing the centralized CBF. In addition, decomposing the centralized CBF allows each agent to solve the optimization problem individually based on their local information in a distributed fashion, and therefore reduces computation costs [58, 114, 79].

Another way to construct a distributed CBF is through a bottom-up approach, e.g., by composing pair-wise CBF. These approaches often encode the constraints of all the pair-wise CBF conditions in a QP to find a feasible control input that can maintain safety with respect to each of the pair-wise CBF [107, 120, 121, 122, 57, 123, 124, 125, 126, 97].

3.2. Predictive safety filter-based shielding

Predictive safety filter (PSF) is another common method of constructing a shield [127], which is also closely related to model predictive shielding (MPS) [128, 129].

For PSF, let $\mathcal{X}_{\text{CI}} \subseteq \mathcal{S}$ denote a subset of the safe set that is *control invariant*, i.e., there exists some controller under which this set is forward invariant. During deployment, at each time step, a constrained optimization

problem is solved that constrains the terminal state within $\mathcal{X}_{\text{CI}}$.

$$\min_{u_k} \quad \|\boldsymbol{\pi}(\mathbf{x}_0) - \mathbf{u}_0\|^2 \tag{10a}$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad \forall k = 0, 1, \ldots, N-1 \tag{10b}$$

$$\mathbf{x}_k \in \mathcal{S}, \quad \forall k = 0, 1, \ldots, N-1 \tag{10c}$$

$$\mathbf{x}_N \in \mathcal{X}_{\text{CI}}. \tag{10d}$$

The output of the safety filter $\boldsymbol{\Pi}$ is then taken as the solution $\mathbf{u}_0$ of (10). The terminal state constraint (10d) helps guarantee recursive feasibility of (10) and, as a result, guarantees infinite-horizon constraint satisfaction.

In MPS, suboptimality of the underlying constrained optimization problem is traded for substantially lowered computational costs. Instead of solving the potentially nonlinear optimization problem (10) at each time-step, a merely feasible solution is found. Let $\boldsymbol{\pi}_{\text{backup}}$ denote a *backup* policy designed to bring the system inside a set $\mathcal{X}_{\text{FI}} \subseteq \mathcal{S}$ that is forward invariant under $\boldsymbol{\pi}_{\text{backup}}$. Then, MPS chooses $\mathbf{u}_0 \in \{\boldsymbol{\pi}(\mathbf{x}_0), \boldsymbol{\pi}_{\text{backup}}(\mathbf{x}_0)\}$ and assigns $\mathbf{u}_k = \boldsymbol{\pi}(\mathbf{x}_k)$ for $k > 0$ [128, 129]. If taking $\mathbf{u}_0 = \boldsymbol{\pi}(\mathbf{x}_0)$ does not lead to a feasible solution (i.e., $\mathbf{x}_N \in \mathcal{X}_{\text{FI}}$), then $\mathbf{u}_0$ is taken to be $\boldsymbol{\pi}_{\text{backup}}(\mathbf{x}_0)$.

MPS is extended to the multi-agent case in [46], where safety under $\boldsymbol{\pi}_{\text{backup}}$ is considered agent-wise as opposed to for the entire MAS. This helps avoid suboptimal cases where all agents are forced to use $\boldsymbol{\pi}_{\text{backup}}$ even when only a single agent is not safe under $\boldsymbol{\pi}$. However, [46] requires a centralized node to compute the MPS, and hence may have challenges scaling to a larger number of agents.

In [17], a PSF-like shield is used without the terminal state constraint (10d) for the *joint* MAS. Linear dynamics and linear constraints are considered, which, along with no terminal constraints, allows for (10) to be solved efficiently as a QP. Although removing the terminal state constraint also removes recursive feasibility guarantees, infeasibility was not reported in [17] during hardware experiments.

Finally, in [80], a distributed method of PSF for linear systems with linear safety constraints and bounded disturbances is introduced. The disturbances are handled using a robust distributed MPC technique that uses tube MPC [130], and a distributed negotiation procedure is introduced to allow agents to trade safety margins with neighbors.

3.3. Hamilton-Jacobi-based shielding

The HJ methods are a class of optimal control tools for finding the reachable set of a dynamical system under worst-case disturbances; [131] provides a good introduction to this field. Most HJ methods proceed by solving a partial differential inequation for a scalar field over the joint state. This scalar field is known as the *HJ value function* and its zero superlevel set is the control invariant set. The HJ value function also defines a controller that renders the system safe. A common HJ shielding strategy is to use this controller as a shield that only activates if the value function gets too close to zero [131]. This shielding method can lead to undesirable bang-bang control behavior. Other formulations produce smoother HJ shielding controllers [132]. The HJ-based methods have also been used to guide the learning of CBF controllers [133].

A classic weakness of HJ methods is that they require solving a partial differential inequation over the state space of the system, and so the computational and memory requirements scale exponentially with the dimension of the state of the system [131]. The memory requirements can be reduced by using function approximations, such as neural networks, to represent the HJ value function [134, 135]. However, this dependence on dimensionality nevertheless prevents HJ methods from being applied to the joint state of MAS. Instead, these methods factor the MAS into pairs of agents and then solve a pairwise HJ problem [81].

3.4. Automata-based shielding

Shielding has also been applied using tools from the field of *formal methods*, where safety requirements are defined using linear temporal logic [136]. Shielding for safe learning-based control using automata was introduced in a single agent case in [137], borrowing ideas from [138], where a safety game [139] is solved to compute a set of states from where safety can be preserved.

In [72], this is extended to the multi-agent case. Instead of constructing a single shield that monitors all agents, the state-space is decomposed into multiple pieces, and a shield is constructed for each piece. The authors show that this allows the algorithm to scale from two to four agents on a grid world. This work is extended in [84], where this decomposition occurs dynamically. In [77], a decentralized shield is constructed, improving scalability.

However, the shield synthesis tools used in these works require a finite abstraction of the state and control spaces [138, 137] and scale exponentially

with the abstraction size [140]. This can lead to conservative behavior when coarse abstractions are used [137].

3.5. Heuristic Projection

Finally, there are shielding methods that do not provide formal safety guarantees but rather act as heuristics. In [85], the heuristic approach from [141] is used as a shield in a MARL framework. Here, the safety constraints are linearized, and it is assumed that only a single constraint is active at each time, giving rise to a closed-form solution that can be computed easily. In [73], the velocity obstacle approach [142] is used as a shield. Here, agents' velocities inside the velocity obstacle are projected back to the safe set. For a dynamic obstacle moving with constant velocity, a velocity obstacle defines the set of velocities of the agent that results in a collision with the dynamic obstacle [142]. When other agents are modeled as dynamic obstacles, the constant velocity assumption may not hold, but variants that make similar assumptions have been used successfully in practice [143].

The tradeoff between improved ease of use and computational cost is that these methods do not have the same safety guarantees as the previous categories of shields.

4. Learning control barrier functions for MAS

Shielding is a powerful technique to guarantee the safety of the MAS. However, hand-crafting a shield is generally difficult, requires domain expertise, and can be done for a relatively small class of problems. Furthermore, it is computationally heavy to find a shield through optimization, especially for large-scale MAS with complex dynamics. To address this problem, there has been a lot of development on using machine learning to find a shield and use it as a guide for controller synthesis. Most of these works focus on a specific kind of shield, namely, CBFs [82], and tend to compute a CBF and a safe control policy simultaneously. In this manner, the computed control policy is encouraged to satisfy the CBF constraints and can be used for satisfying the safety requirements. In this section, we review approaches that learn a CBF and a control policy, for both centralized and distributed settings (see Table 4 for an overview of learning CBF methods).

4.1. Centralized CBF

There is a plethora of work on learning centralized CBF for safety [144, 145, 146, 147, 148, 153, 154, 155, 156, 151]. The general idea of these

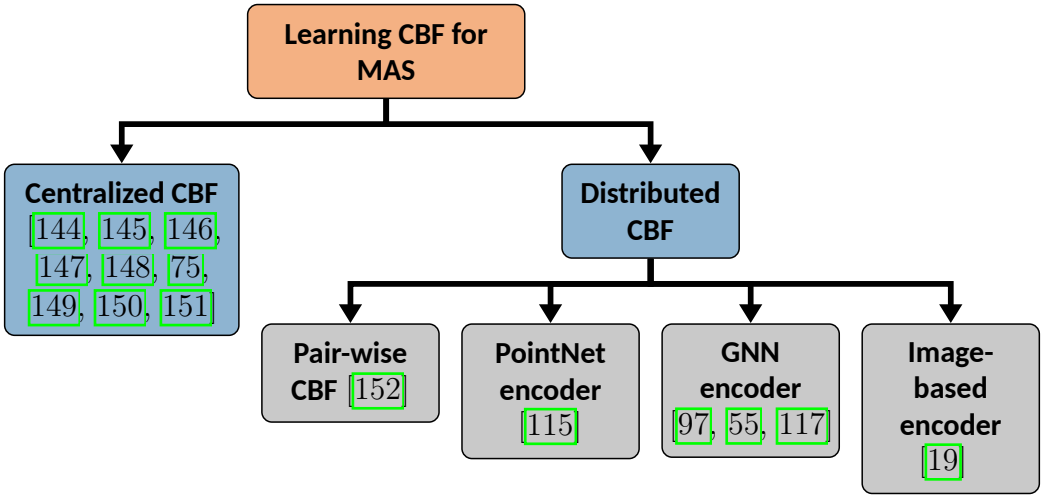

Figure 4: Overview of learning CBF for MAS

approaches is to use *self-supervised learning* for learning a common CBF for the entire MAS [82]. To this end, first, a centralized CBF $B_\theta : \mathcal{X} \to \mathbb{R}$ and a centralized control policy $\boldsymbol{\pi}_\phi : \mathcal{X} \to \mathcal{U}$ are parameterized using Neural Networks (NNs) with parameter θ and ϕ, respectively. Next, a loss function is designed to map the CBF constraint (6) to a penalty term:

$$\mathcal{L}_{\text{deriv}}(\theta, \phi) = \frac{1}{N_{\text{sample}}} \sum_{\mathbf{x} \in \mathcal{X}} \left[-\frac{\partial B_\theta}{\partial \mathbf{x}} \mathbf{F}(\mathbf{x}, \boldsymbol{\pi}_\phi(\mathbf{x})) - \alpha\left(B_\theta(\mathbf{x})\right) \right]^+, \tag{11}$$

where $[\cdot]^+ = \max(\cdot, 0)$ denotes the `ReLU` function, N_{sample} is the total number of state samples collected, and α is an extended class-$\mathcal{K}$ function[9]. To render the safety of the system, following Remark 1, it is essential that the 0-superlevel set of B_θ, denoted as $\mathcal{C}_\theta$, is a subset of the *known* safe set $\mathcal{S}$, i.e., $\mathcal{C}_\theta \subset \mathcal{S}$. However, ensuring such a condition is not straightforward during the learning process.

Therefore, one cannot naively sample from the safe space and constrain the value of B_θ on these samples to be non-negative. To design self-supervised learning losses, most works consider an alternative way: making B_θ negative everywhere in the unsafe space $\mathcal{X} \setminus \mathcal{S}$ [145, 146]. Such a loss function can be

[9] In practice, a linear function $\alpha(y) := \alpha y$ is chosen, for some $\alpha > 0$.

readily defined as

$$\mathcal{L}_{\text{unsafe}}(\theta) = \frac{1}{N_{\text{unsafe}}} \sum_{\mathbf{x} \in \mathcal{X} \setminus \mathcal{S}} [B_\theta(\mathbf{x})]^+ , \tag{12}$$

where N_{unsafe} is the number of state samples in the unsafe space. However, if a loss of the form $\mathcal{L}_{\text{unsafe}}(\theta) + \mathcal{L}_{\text{deriv}}(\theta, \phi)$ is used, since there are only *negative* samples, the learned policy can easily converge to a sub-optimal solution where the forward-invariant set $\mathcal{C}_\theta$ is relatively very small, if not empty. Hence, it is essential that *positive* samples from the safe set are also used in learning the barrier function. Naively using *positive* samples from the safe set $\mathcal{S}$ and *negative samples* from the unsafe set $\mathcal{X} \setminus \mathcal{S}$ does not work since a point being safe (i.e., $x \in \mathcal{S}$) does not imply that the dynamical system can remain safe in the future if it passes through this point. Hence, the *positive* samples must be collected from a control invariant set $\mathcal{C} \subset \mathcal{S}$ so that the system can be guaranteed to remain safe at all future times if it passes through these samples. Based on this, most work on CBF learning e.g., [145, 146, 149, 150] consider an additional loss term:

$$\mathcal{L}_{\text{safe}}(\theta) = \frac{1}{N_{\mathcal{X}_\mathcal{C}}} \sum_{\mathbf{x} \in \mathcal{X}_\mathcal{C}} [-B_\theta(\mathbf{x})]^+ , \tag{13}$$

where $\mathcal{X}_\mathcal{C} \subset \mathcal{S}$ is an approximation of the *unknown* ground truth forward invariant set $\mathcal{C}$, and $N_{\mathcal{X}_\mathcal{C}}$ is the number of state samples in this set, which act as *positive* samples. Since the ground truth forward invariant set $\mathcal{C}$ is computationally expensive to find, researchers generally approximate it with the set $\mathcal{X}_\mathcal{C}$ through various methods, such as using the set of initial conditions if it is known to be part of the forward-invariant set [146], using distance-based heuristic methods [145, 150], or approximating it with a look-forward mechanism [117].

Furthermore, to avoid learning a *flat* CBF whose value is close to 0 over the entire state space, many works also add a small margin $\nu > 0$ in the loss terms [75]. Some works also consider a reference behavior cloning controller $\boldsymbol{\pi}_{\text{BC}}$ for the liveness property, such as reaching a goal location. As a result, another loss term

$$\mathcal{L}_{\text{ctrl}} = \frac{1}{N} \sum_{\mathbf{x} \in \mathcal{X}} \|\boldsymbol{\pi}_\phi(\mathbf{x}) - \boldsymbol{\pi}_{\text{BC}}(\mathbf{x})\|^2,$$

is added so that the learned controller is as close to the behavior cloning controller as possible [75]. Most of the works use a nominal controller $\boldsymbol{\pi}_{\text{nom}}$ as the behavior cloning controller, which only considers the liveness property without the purpose of collision [75, 115, 55]. However, the safety and the liveness properties encoded here via the means of different loss terms compete with each other in learning and often lead to a sub-optimal solution which either has a high performance with poor safety rate (learned controller close to the nominal controller) or high safety rate with poor performance [55]. Recently, [117] proposes to use the controller solved from CBF-QP (8) as the behavior cloning controller, which solves the competition problem between safety and liveness.

Based on these individual loss terms, the parameterized CBF B_θ and the control policy $\boldsymbol{\pi}_\phi$ are trained using the loss function defined as

$$\mathcal{L}_{\text{CBF}}(\theta, \phi) = \mathcal{L}_{\text{deriv}}(\theta, \phi) + \mathcal{L}_{\text{unsafe}}(\theta) + \mathcal{L}_{\text{safe}}(\theta) + \mathcal{L}_{\text{ctrl}}(\phi). \tag{14}$$

Upon convergence, a CBF and a safe control policy are obtained. The training data for such learning methods are either collected by random sampling in the state space [146] or from simulated trajectories [150, 149]. These approaches are generalizable to a large class of dynamics and relatively high-dimensional systems in contrast to the limited applicability of non-learning approaches. Also, these methods propose to learn a safe control policy along with the CBF. As a result, there is no need to solve the CBF-QP (8) during the execution, enabling them for real-time implementation. However, since the learned CBF is a neural network, it is difficult, if not impossible, to verify that the learned CBF candidate satisfies the CBF conditions (6) *everywhere* in the state space. In addition, the forward invariant set $\mathcal{X}_\mathcal{C}$ used in training is not easy to find. Moreover, as a centralized approach, it still needs global information and hence, it is not scalable for large-scale MAS [115, 55].

The data requirements of learned CBFs vary depending on the amount of dynamics information available. There is some work, albeit limited, on learning for robotics systems with safety constraints using hardware data [157, 158, 159]. If the system dynamics are known, then the CBF can be trained entirely in simulation with samples from the safe and unsafe regions. If the dynamics are unknown, then the derivative loss $\mathcal{L}_{deriv}$ in Eq. (11) must be computed using state-action pairs collected on hardware, where samples are usually only available from the safe region. In this case, the safety guarantees provided by the CBF are weakened. If the CBF derivative

condition is only satisfied in the safe set, the system is still guaranteed to remain safe when starting in the safe region, but we are no longer guaranteed to converge to the safe set if we start in the unsafe set. Note that $\mathcal{L}_{safe}$ and $\mathcal{L}_{unsafe}$ only require evaluating the learned CBF on a state, and hence, they can be evaluated on unsafe states in simulation without requiring unsafe data.

4.2. Distributed CBF

In order to eliminate the need for global information and address the limited scalability of centralized CBF learning methods, researchers are now focusing on distributed CBF approaches[115, 152, 97, 55]. These methods assume that agents have access only to local observations and communication within their immediate vicinity. Most of the works suppose that the agents are identical and share the same CBF and control policy. Thus, they use NNs to parameterize *one* CBF B_θ and *one* control policy π_ϕ that can be used for each agent [115, 97, 55], or each pair of agents [152], and use a similar procedure as discussed in Section 4.1 for learning them. Different from learning centralized CBF works, distributed CBF learning works often adopt the centralized training and distributed execution (CTDE) framework. During centralized training, the agents are trained jointly and the loss of agent i's CBF can be backpropagated to its neighbors $j \in \mathcal{N}_i$ so that the distributed CBF conditions (9) are satisfied for all the agents [55]. During distributed execution, the agents apply the learned controller which only uses the local observations to obtain control inputs.

In contrast to the centralized CBF, where the central computation node has global observation, each agent only has local observation in distributed CBF. Different methods have been used to encode the local observation. For example, [115] uses a PointNet [160] so that the observation encoding is permutation-invariant to the observed agents. The recent work [55] proposes to use graph neural networks (GNNs) with attention mechanism [161] to encode the local observation. It utilizes the fact that GNNs with attention can handle a changing number of neighbors. The attention mechanism also addresses the problem of abrupt changes in the CBF value when an agent enters or leaves the sensing region of another agent. For image-based observations, convolutional neural networks (CNNs) and Long Short-Term Memory (LSTM) are used for encoding [19].

Since learning a CBF for a large-scale MAS requires sampling from a large state space, it is not computationally tractable to explore the complete

state space during learning and hence, the safety rate during execution is generally very low. To this end, an online policy refinement technique is applied during execution to obtain better safety performance [115, 55]. The online policy refinement step adds a runtime gradient descent process to update the NN controller during execution. At any time step, if the learned control input does not satisfy the CBF descent condition, then the residue $\delta = [-\dot{B} - \alpha(B)]^+$ is computed, and gradient descent is used to update the learned control to minimize this residue. This is a distributed approach since computing $\dot{B}$ needs the knowledge of the control inputs of the neighboring agents also.

Trained with a few tens of agents, the learned distributed CBF has demonstrated impressive generalizability in very large-scale systems constituting thousands of agents [115, 55]. These approaches are also capable of using realistic and noisy LiDAR observation [55] or image-based pixels [19], instead of assuming that the *actual* relative states are available. Moreover, some approaches also consider contracts among agents in MAS, e.g., agents have different responsibilities for avoiding collisions, and learn this contract [119]. While these methods have several advantages as listed above, the learned CBF is hard to verify for correctness, and cannot provide formal guarantees on the safety of the MAS. Another challenge for these myopic distributed methods is deadlocks[162, 163], thereby compromising on liveness properties. In particular, as proved in [163], Lyapunov-CBF-QP methods induce undesirable equilibrium points which lead to deadlocks in MAS, where none of the agents can move with the CBF-based policy without violating the safety of the MAS.

5. Safe multi-agent reinforcement learning

Instead of delegating satisfaction of safety constraints to shielding-based methods or a learned CBF, one can also directly learn a policy to satisfy the safety constraints. One popular method of learning such a control policy is reinforcement learning (RL). In recent years, many of RL's biggest successes have been in its ability to play multi-agent games ranging from two-agent zero-sum games such as Go [164, 165] and Shogi [166], multi-agent zero-sum games such as Poker and Siplomacy, and team-based games such as Starcraft [167], Dota [168], Honor of Kings[169] and Football[30].

Despite these successes, there have been relatively few works that explicitly examine safety in Multi-Agent RL (MARL). The numerous approaches

to both single-agent and multi-agent reinforcement learning address safety specifications by either terminating the episode when the safety specification is not met (e.g., in [170]) or by including reward terms that discourage the *unsafe* behavior (e.g., in [170, 171, 172]). While these approaches do not provide safety guarantees and can require reward function tuning, they are more popular in comparison to methods that explicitly address safety constraints.

Flavors of safety in RL In the single-agent RL setting, a Constrained Markov Decision Process (CMDP) [173] is the most popular problem formulation that additionally captures safety constraints. A MDP (Markov Decision Process) is defined as the tuple $(\mathcal{X}, \mathcal{U}, \mathbb{P}, r, \rho_0, \gamma)$, where $\mathbb{P}$ denotes the state transition probability, $r : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \to \mathbb{R}$ denotes the reward function, ρ_0 denotes the starting state distribution, and γ denotes the discount factor [174]. In the context of MAS, the reward function may encode liveness properties as in 2.2.2, such as goal reaching [175, 46, 72]. A CMDP $(\mathcal{X}, \mathcal{U}, \mathbb{P}, r, \rho_0, \gamma, C, d)$ extends a MDP by also considering a constraint function $C : \mathcal{X} \to \mathbb{R}$ and constraint bound $d \in \mathbb{R}$. A CMDP seeks a policy π that satisfies the *average cost* constraints

$$\mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k C(x_k) \right] \leq d. \tag{15}$$

While average cost constraints (15) are different from the safety constraints in Definition 1, the average cost constraints can be viewed as *chance constraints* under the *discounted* occupation measure q_π of π [173, Chapter 3], where C is taken to be an indicator function $x \mapsto \mathbb{1}\{x \notin \mathcal{S}\}$ and $d \in [0, 1]$ denotes the probability threshold, since

$$\mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k \mathbb{1}\{x_k \notin \mathcal{S}\} \right] = \mathbb{E}_{q_\pi} \Big[\mathbb{1}\{x \notin \mathcal{S}\} \Big] = \Pr(x \notin \mathcal{S}). \tag{16}$$

On the other hand, for general choices of C and d, this becomes a different notion of safety than Definition 1, and the optimal solution of the CMDP could result in exiting the safe set $\mathcal{S}$. Single-agent RL methods that explicitly solve the CMDP label themselves as constrained RL or safe RL (e.g., [173, 31, 176]) and are commonly tested on benchmarks such as safety gym [177].

Another notion of safety is when no constraint violation is allowed at any *any* time-step k, i.e.,

$$\max_{k \geq 0} C(x_k) \leq d \tag{17}$$

This is known as a peak constraint [178, 179] or state-wise safety [180] in the Safe RL literature, and aligns with the notion of safety considered in this paper, where the safe set $\mathcal{S}$ is defined as

$$\mathcal{S} \coloneqq \{x \in \mathcal{X} \mid C(x) \leq d\}. \tag{18}$$

While peak constraints are not as common for constrained RL methods, some works tackle this problem [134, 176, 181] (see [180] for a recent survey of some methods). It can be related to the average cost case (15) by taking $\tilde{C}_k \coloneqq \max(0, C_k - d)$ and $\tilde{d} = 0$ [181], yielding the constraints $\tilde{C}_k \leq \tilde{d}$ given as

$$\sum_{k=0}^{\infty} \gamma^k \max(0, C_k - d) \leq 0. \tag{19}$$

For both types of constraints, taking smaller values of d results in a stricter constraint on the total accrued cost and hence a lower total reward, while higher values of d loosen the constraint and allow higher rewards.

Safe MARL Given that solving the CMDP problem and obtaining policies that successfully satisfy the safety constraints is already challenging in the single-agent case, it is even more challenging in the MAS setting. Consequently, there have been relatively fewer works that tackle the problem of safe MARL. This is noted in many surveys on MARL, which mention safe MARL to be a direction that is relatively unexplored [26, 24, 31, 182]. Nevertheless, recent years have seen an increasing number of works that tackle this challenging problem. We provide an overview of existing methods in Figure 5 , which we describe in more detail in the coming subsections.

5.1. Unconstrained MARL

Early works that approached the problem of safety for MARL focused on navigation problems and collision avoidance [175, 183, 18, 184], where safety is achieved by either a sparse collision penalty [191], or a shaped reward term that penalizes small distances to obstacles and neighboring agents [175, 183, 18, 184]. However, the satisfaction of collision avoidance constraints is not necessarily guaranteed by either the final policy or even the optimal policy [192]. Consequently, while these methods report 100% safety rates empirically when there are fewer agents, safety violations occur when tested with a larger number of agents (e.g., $> 3\%$ for 8-agents in [18], $> 3\%$ for 20-agents in [191]). These trends are also similar for liveness properties. In [18], although the use of reward to encourage goal-reaching is sufficient for all agents to

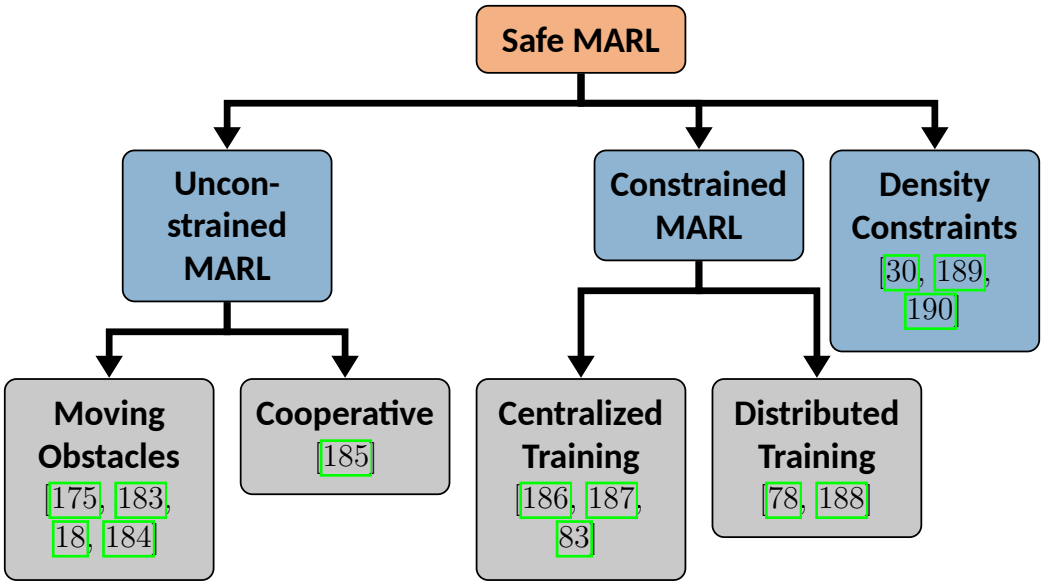

Figure 5: Overview of Safe MARL-based approaches

eventually reach their goal when there are less than 3 agents, the percentage of deadlocks increases to 1.6% for 10 agents. Improving satisfaction of the liveness properties may require alternate approaches such as imitation learning of Multi-Agent Path Finding (MAPF) algorithms [193, 194], where success rates are close to 100% even for up to 128 agents.

While there exists some finite scale for the penalty such that the optimal policy is guaranteed to satisfy the constraints [192], too large of a penalty term often results in poorer performance empirically [195]. As noted in [195], this can be explained by larger penalties resulting in larger variances in the reward which ultimately results in poorer optimization performance.

5.2. Constrained MARL

In contrast to unconstrained MARL methods that penalize safety violations in the reward term and then solve the resulting unconstrained problem, constrained MARL methods explicitly solve the CMDP problem. For the single-agent case, prominent methods for solving CMDPs include primal-dual methods using Lagrange multipliers [196, 197] and via trust-region-based approaches [198]. These methods provide guarantees either in the form of asymptotic convergence guarantees to the optimal (safe) solution [196, 197] using stochastic approximation theory [199, 200], or recursive feasibility of

intermediate policies [198, 201] using ideas from trust region optimization [202]. The survey [31] provides an in-depth overview of the different methods of solving safety-constrained single-agent RL.

MARL algorithms can be broadly divided into two paradigms: centralized and distributed training algorithms [29]. The same holds for their extensions to consider safety constraints.

5.2.1. Centralized training

During centralized training, agent policies are updated at a central node, where information in addition to each agent's local observations is used to update each agent's policies. This is the dominant paradigm for unconstrained MARL [29] and is referred to as Centralized Training Decentralized Execution (CTDE).

One of the first safe CTDE methods to be proposed is in [83], where the authors combine Constrained Policy Optimization (CPO) [198], a method for solving CMDPs, with Heterogeneous-Agent Trust Region Policy Optimisation (HATRPO), a MARL method that enjoys a theoretically-justified monotonic improvement guarantee [203]. Theoretical analysis guarantees monotonic improvement in reward while theoretically satisfying safety constraints during each iteration, assuming that the initial policy is feasible, the value functions are known and a trust-region optimization problem can be solved exactly. However, neither of these assumptions is guaranteed in the method implemented in practice due to approximation errors in the value function and a quadratic approximation of the trust-region problem [83, 203].

Another early safe CTDE method is CMIX [186], which extends the value function factorization method QMIX [204] to additionally consider both average constraints and peak constraints. However, no theoretical analysis of the convergence of the proposed algorithm is given. [186, 187, 83]

5.2.2. Distributed training

In distributed training methods, each agent has a private reward function, policy updates occur locally for each agent, and communication between agents is used to arrive at a policy that minimizes the total reward subject to safety constraints [78]. In this distributed setting where the reward and constraints are private and each agent has a different policy when the algorithm has not converged, it is not possible to evaluate the performance of each agent's policy. Consequently, these approaches are based on primal-dual

optimization and provide asymptotic convergence guarantees to local optima [78], but are unable to provide any safety guarantees before convergence.

5.3. Density-based approaches

Finally, a vastly different approach to safe MARL looks at the case when the agents are indistinguishable, the number of agents is very large and applies the mean-field approximation, where they look at the limit as the number of agents goes to infinity, and the quantity of interest is instead the density of the overall swarm of agents [205, 206]. As the concept of individual agents is gone, both inter-agent and agent-obstacle constraints are replaced by density constraints [207, 208]. Navigation problems when applying the mean-field approximation can be viewed as an optimal transport problem [208]. When also considering safety constraints via state-dependent costs, the problem becomes a mean field game with distributional boundary constraints that can be solved using RL techniques [208].

For example, [207, 208] consider a navigation problem with obstacles, where the density of the multi-agent swarm at the obstacles is constrained to be zero for a given initial and final distribution of agents. In [208], a penalty on the density of agents is also included to incentivize agents to spread out more, and consequently reduce the risk of inter-agent collisions.

6. Safety verification for learning-enabled MAS

Once a control policy has been learned, it must be checked for correctness before it can be deployed, particularly in safety-critical control contexts. This is particularly true for control policies represented using difficult-to-interpret models such as NNs. This section reviews methods for checking the safety properties of a learned controller or shield for MAS. Broadly speaking, these methods can be organized as shown in Fig. 6, where the primary trade-off is between the ability to provide formal guarantees and the ability to scale to practically-sized problems. Fig. 6 also highlights the specific challenges of MAS verification and summarizes existing approaches to resolving these challenges, as well as open problems regarding the limitations of these methods.

We begin by reviewing the methods from Fig. 6 with reference to the extensive literature for single-agent verification, before highlighting the specific challenges that arise in the multi-agent setting. We then review methods

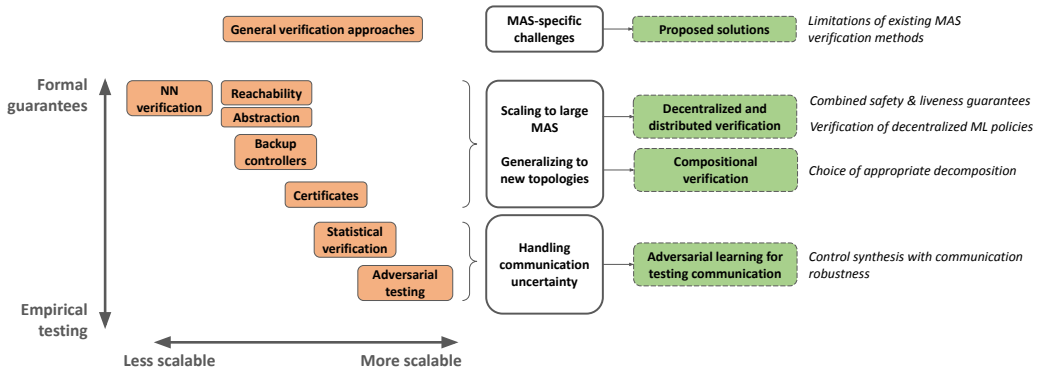

Figure 6: An overview of verification methods, comparing the ability to provide formal guarantees with the ability to scale in practice.

for addressing each of these challenges, concluding with a discussion of open problems.

6.1. Review of single-agent verification tools

Several excellent surveys for centralized verification provide a good starting point for readers interested in a broad introduction to the field; we will review these surveys here, and then devote most of our attention to this survey to the challenges that distinguish multi-agent verification from the centralized case.

The survey [209] covers search- and optimization-based methods for black-box autonomous systems. These methods can sometimes provide formal guarantees, depending on the completeness of the underlying optimizer or search algorithm; for example, some stochastic global optimization methods enjoy asymptotic completeness guarantees under certain assumptions, which may be used to provide formal guarantees, but most black-box optimization methods are best suited for empirical testing. In this survey, we discuss the challenges in scaling these methods to large-scale MAS, as well as how search- and optimization-based methods can be extended to consider issues that are specific to MAS, such as communication noise and cybersecurity.

Centralized reachability analysis using Hamilton-Jacobi (HJ) methods are covered in the survey [131], which are specialized for checking set invariance properties but can provide formal guarantees. The survey [131] discusses centralized verification of MAS (i.e. collapsing the MAS into a single dynamical

system) and verification of pairwise safety (i.e. checking for inter-agent collision avoidance), but they do not consider MAS safety more generally. Other methods of centralized reachability analysis (e.g., set propagation), have been extensively covered in previous surveys [210, 211, 212]. These methods use sets to over-approximate the reachable set of a dynamic system and can be used to provide safety guarantees. However, these surveys do not focus on the topics of learned controllers or the reachability of MAS. We restrict our discussion to decomposition-based approaches to scaling HJ methods, along with other reachability analysis tools, for MAS.

Another recent review article [82] provides a survey of neural certificates, but they focus largely on the centralized case. We extend the discussion of certificates to consider decentralized, distributed, and compositional approaches checking to certificates. Due to the extensive coverage of certificate learning in Section 4, in this section we only discuss the challenges involved in checking the validity of these certificates in the multi-agent case.

In [213], the authors provide a survey of techniques for verifying input-output properties of neural networks, some of which have been applied to verify the soundness of shields like barrier and Lyapunov functions [214, 215]. Due to the extreme fundamental complexity of neural network verification (an NP-complete problem [213]) and several open technical challenges that we discuss in the sequel, these methods have yet to be applied to MAS.

Similarly, other surveys cover verification for safe learning and control [32] and autonomous aerial systems [216]; while these provide a good overview of their respective fields, they do not specifically address MAS.

6.2. Challenges of MAS verification

While many of the concepts from centralized verification extend to the multi-agent case, several challenges are unique to MAS. We discuss these challenges here, and the rest of this section is devoted to reviewing proposed methods for addressing these issues.

6.2.1. Scalability to high dimensions

The most immediate challenge in verifying learned control systems for MAS is the scale. Formal verification using methods like HJ [131] and abstraction [217] are notoriously difficult to scale to systems with large state dimensions, so verifying a MAS by naively concatenating the states of individual agents to form a centralized verification problem is generally not tractable. Similarly, neural network verification is generally intractable for

networks with more than a few hundred neurons [213], so the formal guarantees that might be derived using these methods are hard to apply in practice without additional decomposition of the problem.

Even for empirical testing methods, such as those relying on stochastic optimization, it can be challenging to efficiently explore high-dimensional space. The challenge of scalability has led to a range of specialized techniques for MAS and other large-scale systems, including decentralized certificates, distributed search and optimization, and compositional verification methods, which we discuss in the following section.

6.2.2. Handling changing system topology

The second specific challenge for verifying MAS is that, unlike single-agent systems, MAS can be dynamically reconfigured at runtime. For example, the connectivity of agents might change during execution, or the exact number of agents in the system might be uncertain at test time. As a result, to be most useful for MAS, verification methods must be robust to both the number and topology of agents in the system; in particular, formal guarantees should generalize to different system topology, and empirical testing should achieve sufficient coverage of the range of topology that we expect to see at runtime. In the following, we survey several methods that meet this requirement, such as compositional approaches and adversarial testing, and we identify several important open questions in this area.

6.2.3. Handling communication delay and error

An additional unique feature of MAS verification is that there is a class of disturbances — those affecting inter-agent communication — that are typically not present in single-agent settings, and thus typically not considered by single-agent verification methods [32, 209]. Communication disturbances come in many forms, including both naturally occurring factors like communication delay and noisy or lossy communication and adversarial effects like non-cooperative or malicious agents that can send arbitrary messages to other agents (this latter category is important in considering the cybersecurity of MAS).

6.3. Decentralized and distributed verification

To address the first two challenges (scalability and generalizing to changing system configurations), a natural question is how centralized verification

algorithms can be adapted to the distributed or decentralized setting. Decentralized [115, 114] and distributed [55] barrier certificates are one example of this approach, which we discuss extensively in Sections 3 and 4. Other examples include decentralized reachability analysis, e.g. only considering pairwise interactions [131, 218], and distributed optimization to guarantee safety via a multi-agent safe model-predictive control [80].

There are several important considerations for decentralized verification. First, it is not possible to verify some MAS safety and liveness properties in a fully decentralized setting. For example, obstacle avoidance (involving only individual agents and the environment) can be verified in a decentralized manner, but inter-agent collision avoidance requires considering pairs or small cliques of agents [131, 55, 115]. Other properties are more nuanced; for example, connectivity maintenance can be verified using only pairwise communication if the communication graph topology is fixed (this results in a pairwise maximum distance constraint), but if the topology is allowed to vary then agents must be able to compute eigenvalues of the communication graph Laplacian in a distributed manner [219].

A second consideration concerns the common strategy of guaranteeing safety by constructing a *safety filter* using either barrier certificates [115, 114, 55] or reachability analysis [80], as discussed in Section 3. This safety filter limits the range of actions that individual agents may take and often provides formal safety guarantees, but this limit often reduces the optimality of the filtered policy, and this reduced optimality can be more severe for decentralized safety filters. For example, [220] shows that a centralized safe controller achieves a higher task completion rate than a controller that only considers pairwise interactions.

Finally, the application of existing neural network verification (NNV) tools to decentralized learned control policies or certificates remains an open problem; even if more performant NNV tools are developed that can scale to large policy networks, existing NNV algorithms cannot handle architectures like graph neural networks (GNNs) commonly used in learning for decentralized and distributed control [221].

6.4. Compositional verification methods

An alternative to the distributed and decentralized methods discussed above is a more scalable form of centralized verification where agent-level guarantees are hierarchically composed to yield guarantees for the entire

MAS; this class of approach is known as *compositional verification*. Compositional certificates have been used to prove MAS stability properties by composing individual-agent stability certificates that are either learned [222] or found using sum-of-squares optimization [223]. The benefit of compositional approaches to certificate learning is that they require much less data to train; they can be trained at the individual agent level and then generalized to a large number of agents to provide system-level guarantees [222].

A similar compositional approach has been applied to reachability-based verification, for example using reachability to certify pairwise collision avoidance [131], which is composed to give system-level inter-agent collision avoidance guarantees under certain assumptions about the sparsity of agents, or composing reachable sets for subsystems to certify properties of networked systems [224]. It is important to note that decomposition of system-level properties to the agent level necessarily constrains the space of safe control policies, potentially introducing conservatism; e.g. [220] show that composing pairwise collision-free policies leads to more conservative behavior than a more general N-agent decomposition. An important open question in compositional verification is understanding how the choice of decomposition affects the conservatism (or even feasibility) of the resulting control strategy.

6.5. Handling communication uncertainty

A unique feature of MAS (compared to centralized systems) is that they often rely on communication links between agents to function. In a fully decentralized setting, agents might only require local observations of neighbors (e.g. relative position) without explicit communication, but distributed systems can rely on multiple rounds of message passing to achieve consensus or accomplish a distributed optimization task [11, 12]. If this communication is disrupted, or if adversarial agents inject malicious communication packets, then the safety of the overall MAS can be compromised.

Several surveys consider the problem of communication uncertainty and cybersecurity for MAS; common strategies for ensuring robustness to communication issues include fault detection and isolation (FDI), secure consensus algorithms (e.g. where a non-majority subset of agents can be compromised while still maintaining the integrity of the non-compromised agents [225]), and safety filters that maintain sufficient connectivity of the communication graph [219]. Here, we review how these security methods can be extended using learning-based methods. For example, [226] learns a model-free FDI

policy for single-agent systems with multiple faults; this method can be extended for model-free learning-based FDI in MAS. Other works in this vein include using representation learning to classify the network robustness of MAS [227] or using RL to detect intrusions in a networked system [228] or generate adversarial attacks on MAS [229].

A related communication issue arising in MAS is delay, which in the worst case can prevent the system from achieving consensus or even destabilize the system [230]. Several methods have been proposed to handle communication effects, including delay and uncertainty, in the reinforcement learning literature [231, 232, 233, 234]. However, verifying (either formally or empirically) the robustness of a learned controller to these effects remains a challenging open problem.

7. Open Problems

Given the state of the art reviewed in this survey, a few themes stand out as areas for future work. The rest of this section discusses these themes.

7.1. Combined safety and liveness guarantees

While learning-based methods for MAS have seen immense progress in the past decade, much work is still needed when it comes to safety, provable guarantees, and scalability. In the context of learning, there exists a trade-off between liveness properties, such as goal reaching, and safety properties, such as collision avoidance. It is still an open problem as to how to achieve high safety rates or provable safety guarantees along with high performance or guarantees on deadlock resolutions, especially in partially observable systems and while applying distributed algorithms [115, 55].

7.2. Decomposition

For any single-agent method, one of the main challenges for their extension to MAS is how to perform a suitable decomposition that balances the performance and scalability of the method.

Appropriate choice of decomposition for shielding. For shielding-based approaches, this decomposition has been done via distributed computation techniques to efficiently compute a centralized shielding function [79], factoring the state space [72, 84], distributed decomposition of the safety constraints (i.e., [46]), or performing a completely decentralized factorization of

the shielding function such that the communication is not needed [76, 77]. A major open question with these approaches is to explore how the type of decomposition affects the conservatism of the resulting safety filter and its ability to scale and how different choices of shield type (e.g., PSF, CBF,...) can change this tradeoff between ease of construction, scalability, and conservatism.

Appropriate choice of decomposition for verification. Similar to shielding methods, verification methods also often rely on decomposition. An important open problem discussed in Section 6 is how this choice of decomposition affects the conservatism and completeness of the resulting verification scheme. Furthermore, care must be taken to ensure that the decomposition is valid; for example, if a system is only analyzed pairwise for collision avoidance, how does the verification method account for conflicts between more than two agents?

7.3. Practical issues

Safety under communication uncertainty. Another challenge in designing safe methods for MAS is handling dynamically changing MAS configuration, communication delays, package losses, and adversarial communication. Designing control synthesis and verification tools that are both scalable and robust to time-varying topology as well as communication-related issues remains an open problem, as many existing methods either ignore communication effects or rely on domain-specific information.

Strict requirements of CBF-QP. While using CBF for shielding, a crucial practical issue is the feasibility of the CBF-QP. Real-world systems often have input constraints, e.g., torque limit, acceleration limit, etc. The CBF-QP may become infeasible when input constraints are included. Guaranteeing the feasibility of the CBF-QP is an open problem [57, 235, 236], although it is likely that future work will address these issues.

Safe methods are complex and unpopular in practice. When it comes to RL, one of the main challenges for safety in RL in both the single-agent and multi-agent cases is the tradeoff between the simplicity of the algorithm, practical performance, and safety guarantees. With unconstrained RL, in general, neither the resulting policy after training nor the theoretically optimal policy is guaranteed to satisfy the safety constraints [192, 237]. On the other hand, while some constrained RL methods have convergence and safety guarantees

(e.g., [74]), these methods have more components and thus are more difficult to implement and use in practice as compared to unconstrained variants. As a result, these safe methods are relatively unpopular among practitioners compared to unconstrained methods.

Value of methods without practical safety guarantees. Another challenge for safety in MAS is the question of whether learned CBFs or safe MARL methods should be used instead of shielding-based methods when safety guarantees are desired. Although constrained RL can provide per-iteration safety guarantees, this assumes both an initially feasible policy and access to the true value function, neither of which is guaranteed to hold in practice [83]. Similarly, a learned CBF, if verified to be an actual CBF, inherits the theoretical safety guarantees. However, the training process for learning CBFs does not guarantee that this will always be true. On the other hand, shielding-based methods shift this complexity from the algorithm to the user, who must first construct a valid shielding function to provide provable safety guarantees during both training and deployment. It is not clear whether this tradeoff between scalability and practically relevant safety guarantees will be attractive for safety-critical applications.

8. Conclusions

MAS are ubiquitous in today's world, with potential for applications ranging from robotics to power systems, and there is a large body of literature on control of MAS. However, the safe control design of large-scale robotic MAS is a challenging problem. In this survey, we have reviewed how various learning-based methods have shown promising results in addressing some of the aspects of safe control of MAS, such as the safety guarantees of shielding-based methods, the generalizability of learning CBF methods, and the wide applicability of MARL-based methods. Despite their advantages, no existing method has all the desired properties of being provably safe, scalable, computationally tractable, and implementable to a variety of MAS problems. While certificate learning and safe MARL-based methods can provide theoretical safety guarantees, these guarantees do not hold in practice due to unrealizable assumptions. We have identified a range of open problems covering these concerns, and we hope that our review of the state-of-the-art in this field provides a springboard for further research to address these issues and realize the full potential of safe learning-based control for MAS.

9. Acknowledgements

Funding: This work was supported by the National Science Foundation (NSF) CAREER Award #CCF-2238030 and Air Force Office of Scientific Research (AFOSR) grant FA9550-23-1-0099.

References

[1] Ali Dorri, Salil S Kanhere, and Raja Jurdak. "Multi-Agent Systems: A Survey". In: *IEEE Access* 6 (2018), pp. 28573–28593.

[2] Philipp Ringler, Dogan Keles, and Wolf Fichtner. "Agent-based Modelling and Simulation of Smart Electricity Grids and Markets– A Literature Review". In: *Renewable and Sustainable Energy Reviews* 57 (2016), pp. 205–215.

[3] Sergey Vorotnikov et al. "Multi-Agent Robotic Systems in Collaborative Robotics". In: *Interactive Collaborative Robotics: Third International Conference*. Springer. 2018, pp. 270–279.

[4] Jorge Pena Queralta et al. "Collaborative Multi-Robot Search and Rescue: Planning, Coordination, Perception, and Active Vision". In: *IEEE Access* 8 (2020), pp. 191617–191643.

[5] Liang Wang et al. "Multi-Agent Deep Reinforcement Learning-based Trajectory Planning for Multi-UAV Assisted Mobile Edge Computing". In: *IEEE Transactions on Cognitive Communications and Networking* 7.1 (2020), pp. 73–84.

[6] Jingjing Cui, Yuanwei Liu, and Arumugam Nallanathan. "Multi-Agent Reinforcement Learning-based Resource Allocation for UAV Networks". In: *IEEE Transactions on Wireless Communications* 19.2 (2019), pp. 729–743.

[7] Wei Ren. "Distributed Attitude Alignment in Spacecraft Formation Flying". In: *International Journal of Adaptive Control and Signal Processing* 21.2-3 (2007), pp. 95–113.

[8] Caisheng Wei et al. "Learning-based Adaptive Attitude Control of Spacecraft Formation with Guaranteed Prescribed Performance". In: *IEEE Transactions on Cybernetics* 49.11 (2018), pp. 4004–4016.

[9] Jie Huang et al. “Integrated Planning and Control for Formation Reconfiguration of Multiple Spacecrafts: A Predictive Behavior Control Approach”. In: *Advances in Space Research* 72.6 (2023), pp. 2007–2019.

[10] Oren Salzman and Roni Stern. “Research Challenges and Opportunities in Multi-Agent Path Finding and Multi-Agent Pickup and Delivery Problems”. In: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. 2020, pp. 1711–1715.

[11] Daniel K Molzahn et al. “A Survey of Distributed Optimization and Control Algorithms for Electric Power Systems”. In: *IEEE Transactions on Smart Grid* 8.6 (2017), pp. 2941–2962.

[12] Enrique Espina et al. “Distributed Control Strategies for Microgrids: An Overview”. In: *IEEE Access* 8 (2020), pp. 193412–193448.

[13] Zool Hilmi Ismail, Nohaidda Sariff, and E Gorrostieta Hurtado. “A Survey and Analysis of Cooperative Multi-Agent Robot Systems: Challenges and Directions”. In: *Applications of Mobile Robots* (2018), pp. 8–14.

[14] Lorenzo Canese et al. “Multi-Agent Reinforcement Learning: A Review of Challenges and Applications”. In: *Applied Sciences* 11.11 (2021), p. 4948.

[15] Wei Du and Shifei Ding. “A Survey on Multi-Agent Deep Reinforcement Learning: From the Perspective of Challenges and Applications”. In: *Artificial Intelligence Review* 54 (2021), pp. 3215–3238.

[16] Kingsley Nweye et al. “Real-World Challenges for Multi-Agent Reinforcement Learning in Grid-Interactive Buildings”. In: *Energy and AI* 10 (2022), p. 100202.

[17] Abraham P Vinod et al. “Safe Multi-Agent Motion Planning via Filtered Reinforcement Learning”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 7270–7276.

[18] Michael Everett, Yu Fan Chen, and Jonathan P How. “Motion Planning among Dynamic, Decision-making Agents with Deep Reinforcement Learning”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 3052–3059.

[19] Yuxiang Cui et al. “Learning Observation-Based Certifiable Safe Policy for Decentralized Multi-Robot Navigation”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 5518–5524.

[20] Xueguang Lyu et al. “Contrasting Centralized and Decentralized Critics in Multi-Agent Reinforcement Learning”. In: *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS ’21. International Foundation for Autonomous Agents and Multiagent Systems, 2021, pp. 844–852.

[21] Fei Chen, Wei Ren, et al. “On the Control of Multi-Agent Systems: A Survey”. In: *Foundations and Trends® in Systems and Control* 6.4 (2019), pp. 339–499.

[22] Anam Tahir et al. “Swarms of Unmanned Aerial Vehicles—A Survey”. In: *Journal of Industrial Information Integration* 16 (2019), p. 100106.

[23] Thanh Thi Nguyen, Ngoc Duy Nguyen, and Saeid Nahavandi. “Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications”. In: *IEEE Transactions on Cybernetics* 50.9 (2020), pp. 3826–3839.

[24] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. “Multi-agent Reinforcement Learning: A Selective Overview of Theories and Algorithms”. In: *Handbook of Reinforcement Learning and Control* (2021), pp. 321–384.

[25] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. “Decentralized Multi-Agent Reinforcement Learning with Networked Agents: Recent Advances”. In: *Frontiers of Information Technology & Electronic Engineering* 22.6 (2021), pp. 802–814.

[26] Afshin Oroojlooy and Davood Hajinezhad. “A Review of Cooperative Multi-Agent Deep Reinforcement Learning”. In: *Applied Intelligence* 53.11 (2023), pp. 13677–13722.

[27] Ziyuan Zhou, Guanjun Liu, and Ying Tang. “Multi-Agent Reinforcement Learning: Methods, Applications, Visionary Prospects, and Challenges”. In: *arXiv preprint arXiv:2305.10091* (2023).

[28] Yara Rizk, Mariette Awad, and Edward W Tunstel. “Cooperative Heterogeneous Multi-Robot Systems: A Survey”. In: *ACM Computing Surveys (CSUR)* 52.2 (2019), pp. 1–31.

[29] Sven Gronauer and Klaus Diepold. “Multi-Agent Deep Reinforcement Learning: A Survey”. In: *Artificial Intelligence Review* (2022), pp. 1–49.

[30] Yongshuai Liu, Avishai Halev, and Xin Liu. “Policy Learning with Constraints in Model-Free Reinforcement Learning: A Survey”. In: *The 30th International Joint Conference on Artificial Intelligence (IJCAI)*. 2021.

[31] Shangding Gu et al. “A Review of Safe Reinforcement Learning: Methods, Theory and Applications”. In: *arXiv preprint arXiv:2205.10330* (2022).

[32] Lukas Brunke et al. “Safe Learning in Robotics: From Learning-based Control to Safe Reinforcement Learning”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 5 (2022), pp. 411–444.

[33] Tao Yang et al. “A Survey of Distributed Optimization”. In: *Annual Reviews in Control* 47 (2019), pp. 278–305.

[34] Jing Xie and Chen-Ching Liu. “Multi-Agent Systems and their Applications”. In: *Journal of International Council on Electrical Engineering* 7.1 (2017), pp. 188–197.

[35] Hadi Salman, Elif Ayvali, and Howie Choset. “Multi-Agent Ergodic Coverage with Obstacle Avoidance”. In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 27. 2017, pp. 242–249.

[36] Yan Gao et al. “A Non-Potential Orthogonal Vector Field Method for More Efficient Robot Navigation and Control”. In: *Robotics and Autonomous Systems* 159 (2023), p. 104291.

[37] Qishao Wang et al. “Distributed Model Predictive Control for Linear–Quadratic Performance and Consensus State Optimization of Multiagent Systems”. In: *IEEE Transactions on Cybernetics* 51.6 (2020), pp. 2905–2915.

[38] Cameron Nowzari, Eloy Garcia, and Jorge Cortés. “Event-Triggered Communication and Control of Networked Systems for Multi-Agent Consensus”. In: *Automatica* 105 (2019), pp. 1–27.

[39] Angelia Nedić and Ji Liu. “Distributed Optimization for Control”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 1 (2018), pp. 77–103.

[40] Jianrui Wang et al. “Cooperative and Competitive Multi-Agent Systems: From Optimization to Games”. In: *IEEE/CAA Journal of Automatica Sinica* 9.5 (2022), pp. 763–783.

[41] Aaron D Ames et al. “Control Barrier Function based Quadratic Programs for Safety Critical Systems”. In: *IEEE Transactions on Automatic Control* 62.8 (2016), pp. 3861–3876.

[42] Michael M Zavlanos and George J Pappas. “Distributed Connectivity Control of Mobile Networks”. In: *IEEE Transactions on Robotics* 24.6 (2008), pp. 1416–1428.

[43] Mehran Mesbahi and Magnus Egerstedt. “Graph Theoretic Methods in Multiagent Networks”. In: *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[44] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT press, 2008.

[45] Li Wang, Aaron Ames, and Magnus Egerstedt. “Safety Barrier Certificates for Heterogeneous Multi-Robot Systems”. In: *2016 American control conference (ACC)*. IEEE. 2016, pp. 5213–5218.

[46] Wenbo Zhang, Osbert Bastani, and Vijay Kumar. “MAMPS: Safe Multi-Agent Reinforcement Learning via Model Predictive Shielding”. In: *arXiv preprint arXiv:1910.12639* (2019).

[47] Gökhan M Atınç, Dušan M Stipanović, and Petros G Voulgaris. “A Swarm-based Approach to Dynamic Coverage Control of Multi-Agent Systems”. In: *Automatica* 112 (2020), p. 108637.

[48] Manish Prajapat et al. “Near-Optimal Multi-Agent Learning for Safe Coverage Control”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 14998–15012.

[49] Kaiqing Zhang et al. “Fully Decentralized Multi-Agent Reinforcement Learning with Networked Agents”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5872–5881.

[50] Dawei Sun et al. “Multi-agent motion planning from signal temporal logic specifications”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 3451–3458.

[51] Jingkai Chen et al. “Scalable and safe multi-agent motion planning with nonlinear dynamics and bounded disturbances”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 13. 2021, pp. 11237–11245.

[52] Jorge Cortes et al. “Coverage Control for Mobile Sensing Networks”. In: *IEEE Transactions on Robotics and Automation* 20.2 (2004), pp. 243–255.

[53] Farhad Mehdifar et al. “Prescribed Performance Distance-based Formation Control of Multi-Agent Systems”. In: *Automatica* 119 (2020), p. 109086.

[54] Dimitra Panagou. “A Distributed Feedback Motion Planning Protocol for Multiple Unicycle Agents of Different Classes”. In: *IEEE Transactions on Automatic Control* 62.3 (2016), pp. 1178–1193.

[55] Songyuan Zhang, Kunal Garg, and Chuchu Fan. “Neural Graph Control Barrier Functions Guided Distributed Collision-avoidance Multi-agent Control”. In: *7th Annual Conference on Robot Learning*. 2023.

[56] Lorenzo Sabattini et al. “Distributed Control of Multirobot Systems with Global Connectivity Maintenance”. In: *IEEE Transactions on Robotics* 29.5 (2013), pp. 1326–1332.

[57] Yuxiao Chen, Andrew Singletary, and Aaron D Ames. “Guaranteed Obstacle Avoidance for Multi-Robot Operations with Limited Actuation: A Control Barrier Function Approach”. In: *IEEE Control Systems Letters* 5.1 (2020), pp. 127–132.

[58] Urs Borrmann et al. “Control Barrier Certificates for Safe Swarm Behavior”. In: *IFAC-PapersOnLine* 48.27 (2015), pp. 68–73.

[59] Zhou Xian, Puttichai Lertkultanon, and Quang-Cuong Pham. “Closed-chain Manipulation of Large Objects by Multi-Arm Robotic Systems”. In: *IEEE Robotics and Automation Letters* 2.4 (2017), pp. 1832–1839.

[60] Xianwei Li, Yang Tang, and Hamid Reza Karimi. “Consensus of Multi-Agent Systems via Fully Distributed Event-Triggered Control”. In: *Automatica* 116 (2020), p. 108898.

[61] Wei Ren, Randal W Beard, and Ella M Atkins. “A Survey of Consensus Problems in Multi-Agent Coordination”. In: *American Control Conference*. IEEE. 2005, pp. 1859–1864.

[62] Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. "A Survey of Multi-Agent Formation Control". In: *Automatica* 53 (2015), pp. 424–440.

[63] Lei Xue and Xianghui Cao. "Leader Selection via Supermodular Game for Formation Control in Multiagent Systems". In: *IEEE Transactions on Neural Networks and Learning Systems* 30.12 (2019), pp. 3656–3664.

[64] María Santos and Magnus Egerstedt. "Coverage Control for Multi-Robot Teams with Heterogeneous Sensing Capabilities using Limited Communications". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 5313–5319.

[65] Rupak Majumdar et al. "Symbolic Reach-Avoid Control of Multi-Agent Systems". In: *Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems*. 2021, pp. 209–220.

[66] Kunal Garg and Dimitra Panagou. "Finite-time Estimation and Control for Multi-Aircraft Systems under Wind and Dynamic Obstacles". In: *Journal of Guidance, Control, and Dynamics* 42.7 (2019), pp. 1489–1505.

[67] Antonio Adaldo et al. "Multi-Agent Trajectory Tracking with Self-Triggered Cloud Access". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE. 2016, pp. 2207–2214.

[68] Muhammad Saim et al. "Distributed Average Tracking in Multi-Agent Coordination: Extensions and Experiments". In: *IEEE Systems Journal* 12.3 (2017), pp. 2428–2436.

[69] Ping Xuan and Victor Lesser. "Multi-Agent Policies: From Centralized Ones to Decentralized Ones". In: *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 3*. 2002, pp. 1098–1105.

[70] Maayan Roth, Reid Simmons, and Manuela Veloso. "Decentralized Communication Strategies for Coordinated Multi-Agent Policies". In: *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III: Proceedings from the 2005 International Workshop on Multi-Robot Systems*. Springer. 2005, pp. 93–105.

[71] Kenneth D Frampton, Oliver N Baumann, and Paolo Gardonio. "A Comparison of Decentralized, Distributed, and Centralized Vibro-Acoustic Control". In: *The Journal of the Acoustical Society of America* 128.5 (2010), pp. 2798–2806.

[72] Ingy ElSayed-Aly et al. "Safe Multi-Agent Reinforcement Learning via Shielding". In: *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '21. International Foundation for Autonomous Agents and Multiagent Systems, 2021, pp. 483–491.

[73] Arbaaz Khan et al. "Learning Safe Unlabeled Multi-Robot Planning with Motion Constraints". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 7558–7565.

[74] Shangding Gu et al. "Multi-Agent Constrained Policy Optimisation". In: *arXiv preprint arXiv:2110.02793* (2021).

[75] Charles Dawson et al. "Safe Nonlinear Control using Robust Neural Lyapunov-Barrier Functions". In: *Conference on Robot Learning*. PMLR. 2022, pp. 1724–1735.

[76] Zhiyuan Cai et al. "Safe Multi-Agent Reinforcement Learning through Decentralized Multiple Control Barrier Functions". In: *arXiv preprint arXiv:2103.12553* (2021).

[77] Daniel Melcer, Christopher Amato, and Stavros Tripakis. "Shield Decentralization for Safe Multi-Agent Reinforcement Learning". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 13367–13379.

[78] Songtao Lu et al. "Decentralized Policy Gradient Descent Ascent for Safe Multi-Agent Reinforcement Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 10. 2021, pp. 8767–8775.

[79] Marcus A Pereira et al. "Decentralized Safe Multi-agent Stochastic Optimal Control using Deep FBSDEs and ADMM". In: *Proceedings of Robotics: Science and Systems*. New York City, NY, USA, 2022.

[80] Simon Muntwiler et al. "Distributed Model Predictive Safety Certification for Learning-based Control". In: *IFAC-PapersOnLine* 53.2 (2020). 21st IFAC World Congress, pp. 5258–5265.

[81] Mo Chen et al. “Safe Platooning of Unmanned Aerial Vehicles via Reachability”. In: *2015 54th IEEE Conference on Decision and Control (CDC)*. 2015, pp. 4695–4701.

[82] Charles Dawson, Sicun Gao, and Chuchu Fan. “Safe Control with Learned Certificates: A Survey of Neural Lyapunov, Barrier, and Contraction Methods for Robotics and Control”. In: *IEEE Transactions on Robotics* 39.3 (2023), pp. 1749–1767.

[83] Shangding Gu et al. “Safe Multi-Agent Reinforcement Learning for Multi-Robot Control”. In: *Artificial Intelligence* 319 (2023), p. 103905.

[84] Wenli Xiao, Yiwei Lyu, and John Dolan. “Model-Based Dynamic Shielding for Safe and Efficient Multi-Agent Reinforcement Learning”. In: *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*. AAMAS ’23. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2023, pp. 1587–1596.

[85] Ziyad Sheebaelhamd et al. “Safe Deep Reinforcement Learning for Multi-Agent Systems with Continuous Action Spaces”. In: (2021). ICML Workshop on Reinforcement Learning for Real Life.

[86] Mitio Nagumo. “Über die lage der integralkurven gewöhnlicher differentialgleichungen”. In: *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series* 24 (1942), pp. 551–559.

[87] Franco Blanchini. “Set Invariance in Control”. In: *Automatica* 35.11 (1999), pp. 1747–1767.

[88] Haïm Brezis. “On a Characterization of Flow-Invariant Sets”. In: *Communications on Pure and Applied Mathematics* 23.2 (1970), pp. 261–263.

[89] Eduardo D Sontag. “A Lyapunov-like Characterization of Asymptotic Controllability”. In: *SIAM Journal on Control and Optimization* 21.3 (1983), pp. 462–471.

[90] Aaron D Ames et al. “Rapidly Exponentially Stabilizing Control Lyapunov Functions and Hybrid Zero Dynamics”. In: *IEEE Transactions on Automatic Control* 59.4 (2014), pp. 876–891.

[91] Stephen Prajna and Ali Jadbabaie. "Safety Verification of Hybrid Systems using Barrier Certificates". In: *International Workshop on Hybrid Systems: Computation and Control*. Springer. 2004, pp. 477–492.

[92] Stephen Prajna. "Barrier Certificates for Nonlinear Model Validation". In: *Automatica* 42.1 (2006), pp. 117–126.

[93] Aaron D Ames et al. "Control Barrier Functions: Theory and Applications". In: *18th European Control Conference (ECC)*. IEEE. 2019, pp. 3420–3431.

[94] Peter Wieland and Frank Allgöwer. "Constructive Safety using Control Barrier Functions". In: *IFAC Proceedings Volumes* 40.12 (2007), pp. 462–467.

[95] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. "Control Barrier Function based Quadratic Programs with Application to Adaptive Cruise Control". In: *53rd IEEE Conference on Decision and Control*. IEEE. 2014, pp. 6271–6278.

[96] Keng Peng Tee, Shuzhi Sam Ge, and Eng Hock Tay. "Barrier Lyapunov Functions for the Control of Output-Constrained Nonlinear Systems". In: *Automatica* 45.4 (2009), pp. 918–927.

[97] Chenning Yu, Hongzhan Yu, and Sicun Gao. "Learning Control Admissibility Models with Graph Neural Networks for Multi-Agent Navigation". In: *Conference on Robot Learning*. PMLR. 2023, pp. 934–945.

[98] Guofan Wu and Koushil Sreenath. "Safety-Critical Control of a Planar Quadrotor". In: *2016 American Control Conference (ACC)*. IEEE. 2016, pp. 2252–2258.

[99] Xiangru Xu et al. "Realizing Simultaneous Lane Keeping and Adaptive Speed Regulation on Accessible Mobile Robot Testbeds". In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2017, pp. 1769–1775.

[100] Kunal Garg and Dimitra Panagou. "Control-Lyapunov and Control-Barrier Functions based Quadratic Program for Spatio-Temporal Specifications". In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, pp. 1422–1429.

[101] Kunal Garg and Dimitra Panagou. "Robust Control Barrier and Control Lyapunov Functions with Fixed-time Convergence Guarantees". In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 2292–2297.

[102] Kunal Garg, Ehsan Arabi, and Dimitra Panagou. "Fixed-time Control under Spatiotemporal and Input Constraints: A Quadratic Programming based Approach". In: *Automatica* 141 (2022), p. 110314.

[103] Ji Yin et al. "Shield Model Predictive Path Integral: A Computationally Efficient Robust MPC Method Using Control Barrier Functions". In: *IEEE Robotics and Automation Letters* 8.11 (2023), pp. 7106–7113.

[104] Mukun Tong, Charles Dawson, and Chuchu Fan. "Enforcing Safety for Vision-based Controllers via Control Barrier Functions and Neural Radiance Fields". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 10511–10517.

[105] Kunal Garg et al. "Multi-rate control design under input constraints via fixed-time barrier functions". In: *IEEE Control Systems Letters* 6 (2021), pp. 608–613.

[106] Shao-Chen Hsu, Xiangru Xu, and Aaron D Ames. "Control Barrier Function based Quadratic Programs with Application to Bipedal Robotic Walking". In: *2015 American Control Conference (ACC)*. IEEE. 2015, pp. 4542–4548.

[107] Paul Glotfelter, Jorge Cortés, and Magnus Egerstedt. "Nonsmooth Barrier Functions with Applications to Multi-Robot Systems". In: *IEEE Control Systems Letters* 1.2 (2017), pp. 310–315.

[108] James Usevitch, Kunal Garg, and Dimitra Panagou. "Strong Invariance using Control Barrier Functions: A Clarke Tangent Cone Approach". In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE. 2020, pp. 2044–2049.

[109] Amir Ali Ahmadi and Anirudha Majumdar. "Some Applications of Polynomial Optimization in Operations Research and Real-time Decision Making". In: *Optimization Letters* 10 (2016), pp. 709–729.

[110] Andrew Clark. "Verification and Synthesis of Control Barrier Functions". In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 6105–6112.

[111] Mohit Srinivasan et al. “Extent-Compatible Control Barrier Functions”. In: *Systems & Control Letters* 150 (2021), p. 104895.

[112] Lars Lindemann and Dimos V Dimarogonas. “Control Barrier Functions for Multi-Agent Systems under Conflicting Local Signal Temporal Logic Tasks”. In: *IEEE Control Systems Letters* 3.3 (2019), pp. 757–762.

[113] Dimitra Panagou, Dušan M Stipanovič, and Petros G Voulgaris. “Multi-Objective Control for Multi-Agent Systems using Lyapunov-like Barrier Functions”. In: *52nd IEEE Conference on Decision and Control*. IEEE. 2013, pp. 1478–1483.

[114] Li Wang, Aaron D Ames, and Magnus Egerstedt. “Safety Barrier Certificates for Collisions-free Multirobot Systems”. In: *IEEE Transactions on Robotics* 33.3 (2017), pp. 661–674.

[115] Zengyi Qin et al. “Learning Safe Multi-agent Control with Decentralized Neural Barrier Certificates”. In: *International Conference on Learning Representations*. 2020.

[116] Manao Machida and Masumi Ichien. “Consensus-based Control Barrier Function for Swarm”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 8623–8628.

[117] Songyuan Zhang et al. “GCBF+: A Neural Graph Control Barrier Function Framework for Distributed Safe Multi-Agent Control”. In: *arXiv preprint arXiv:2401.14554* (2024).

[118] Lars Lindemann and Dimos V Dimarogonas. “Barrier Function based Collaborative Control of Multiple Robots under Signal Temporal Logic Tasks”. In: *IEEE Transactions on Control of Network Systems* 7.4 (2020), pp. 1916–1928.

[119] Ryan K. Cosner et al. “Learning Responsibility Allocations for Safe Human-Robot Interaction with Applications to Autonomous Driving”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 9757–9763.

[120] Paul Glotfelter, Jorge Cortés, and Magnus Egerstedt. “Boolean Composability of Constraints and Control Synthesis for Multi-Robot Systems via Nonsmooth Control Barrier Functions”. In: *2018 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2018, pp. 897–902.

[121] Riku Funada et al. “Visual Coverage Control for Teams of Quadcopters via Control Barrier Functions”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 3010–3016.

[122] Wenhao Luo, Wen Sun, and Ashish Kapoor. “Multi-Robot Collision Avoidance under Uncertainty with Probabilistic Safety Barrier Certificates”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 372–383.

[123] Mrdjan Jankovic and Mario Santillo. “Collision Avoidance and Liveness of Multi-Agent Systems with CBF-based Controllers”. In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 6822–6828.

[124] Pravin Mali et al. “Incorporating Prediction in Control Barrier Function based Distributive Multi-Robot Collision Avoidance”. In: *2021 European Control Conference (ECC)*. IEEE. 2021, pp. 2394–2399.

[125] Yifan Hu, Junjie Fu, and Guanghui Wen. “Decentralized Robust Collision-Avoidance for Cooperative Multirobot Systems: A Gaussian Process-Based Control Barrier Function Approach”. In: *IEEE Transactions on Control of Network Systems* 10.2 (2023), pp. 706–717.

[126] Chao Jiang and Yi Guo. “Incorporating Control Barrier Functions in Distributed Model Predictive Control for Multi-Robot Coordinated Control”. In: *IEEE Transactions on Control of Network Systems* (2023).

[127] Kim P Wabersich and Melanie N Zeilinger. “Linear Model Predictive Safety Certification for Learning-based Control”. In: *2018 IEEE Conference on Decision and Control (CDC)*. IEEE. 2018, pp. 7130–7135.

[128] Shuo Li and Osbert Bastani. “Robust Model Predictive Shielding for Safe Reinforcement Learning with Stochastic Dynamics”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 7166–7172.

[129] Osbert Bastani. “Safe Reinforcement Learning with Nonlinear Dynamics via Model Predictive Shielding”. In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 3488–3494.

[130] Christian Conte et al. “Robust Distributed Model Predictive Control of Linear Systems”. In: *2013 European Control Conference (ECC)*. IEEE. 2013, pp. 2764–2769.

[131] Somil Bansal et al. “Hamilton-Jacobi Reachability: A Brief Overview and Recent Advances”. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)* (2017), pp. 2242–2253.

[132] Jason J. Choi et al. “Robust Control Barrier–Value Functions for Safety-Critical Control”. In: *2021 60th IEEE Conference on Decision and Control (CDC)*. Austin, TX, USA: IEEE Press, 2021, pp. 6814–6821.

[133] Sander Tonkens and Sylvia Herbert. “Refining Control Barrier Functions through Hamilton-Jacobi Reachability”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 13355–13362.

[134] Jaime F Fisac et al. “Bridging Hamilton-Jacobi Safety Analysis and Reinforcement Learning”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8550–8556.

[135] Somil Bansal and Claire J Tomlin. “DeepReach: A Deep Learning Approach to High-dimensional Reachability”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 1817–1824.

[136] Amir Pnueli. “The Temporal Logic of Programs”. In: *18th Annual Symposium on Foundations of Computer Science*. ieee. 1977, pp. 46–57.

[137] Mohammed Alshiekh et al. “Safe Reinforcement Learning via Shielding”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[138] Roderick Bloem et al. “Shield Synthesis: Runtime Enforcement for Reactive Systems”. In: *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer. 2015, pp. 533–548.

[139] René Mazala. “Infinite Games”. In: *Automata Logics, and Infinite Games: A Guide to Current Research*. Springer, 2002, pp. 23–38.

[140] Bettina Könighofer et al. “Shield Synthesis”. In: *Formal Methods in System Design* 51 (2017), pp. 332–361.

[141] Gal Dalal et al. “Safe Exploration in Continuous Action Spaces”. In: *arXiv preprint arXiv:1801.08757* (2018).

[142] Paolo Fiorini and Zvi Shiller. “Motion Planning in Dynamic Environments using Velocity Obstacles”. In: *The International Journal of Robotics Research* 17.7 (1998), pp. 760–772.

[143] Jamie Snape et al. “The Hybrid Reciprocal Velocity Obstacle”. In: *IEEE Transactions on Robotics* 27.4 (2011), pp. 696–706.

[144] Matteo Saveriano and Dongheui Lee. “Learning Barrier Functions for Constrained Motion Planning with Dynamical Systems”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 112–119.

[145] Mohit Srinivasan et al. “Synthesis of Control Barrier Functions using a Supervised Machine Learning Approach”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 7139–7145.

[146] Wanxin Jin et al. “Neural Certificates for Safe Control Policies”. In: *arXiv preprint arXiv:2006.08465* (2020).

[147] Alexander Robey et al. “Learning Control Barrier Functions from Expert Demonstrations”. In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE. 2020, pp. 3717–3724.

[148] Andrea Peruffo, Daniele Ahmed, and Alessandro Abate. “Automated and Formal Synthesis of Neural Barrier Certificates for Dynamical Models”. In: *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer. 2021, pp. 370–388.

[149] Bolun Dai, Prashanth Krishnamurthy, and Farshad Khorrami. “Learning a Better Control Barrier Function”. In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE. 2022, pp. 945–950.

[150] Hongzhan Yu et al. “Sequential Neural Barriers for Scalable Dynamic Obstacle Avoidance”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2023.

[151] Oswin So et al. “How to train your neural control barrier function: Learning safety filters for complex input-constrained systems”. In: *2024 International Conference on Robotics and Automation (ICRA)*. IEEE. 2024.

[152] Yue Meng, Zengyi Qin, and Chuchu Fan. "Reactive and Safe Road User Simulations using Neural Barrier Certificates". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 6299–6306.

[153] Lizhi Wang et al. "Learning-Based, Safety and Stability-Certified Microgrid Control". In: *2023 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE. 2023, pp. 1–5.

[154] Wei Xiao et al. "Barriernet: Differentiable Control Barrier Functions for Learning of Safe Robot Control". In: *IEEE Transactions on Robotics* (2023).

[155] Lizhi Wang et al. "Physics-Informed, Safety and Stability Certified Neural Control for Uncertain Networked Microgrids". In: *IEEE Transactions on Smart Grid* (2023).

[156] Zengyi Qin, Dawei Sun, and Chuchu Fan. "Sablas: Learning safe control for black-box dynamical systems". In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 1928–1935.

[157] Ryan Cosner et al. "Safety-aware preference-based learning for safety-critical control". In: *Learning for Dynamics and Control Conference*. PMLR. 2022, pp. 1020–1033.

[158] Dominik Baumann et al. "Gosafe: Globally optimal safe robot learning". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 4452–4458.

[159] Alonso Marco et al. "Robot learning with crash constraints". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1439–1446.

[160] Charles R Qi et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 652–660.

[161] Yujia Li et al. "Graph Matching Networks for Learning the Similarity of Graph Structured Objects". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3835–3845.

[162] Max H Cohen and Calin Belta. "Approximate Optimal Control for Safety-Critical Systems with Control Barrier Functions". In: *2020 59th IEEE conference on decision and control (CDC)*. IEEE. 2020, pp. 2062–2067.

[163] Matheus F Reis, A Pedro Aguiar, and Paulo Tabuada. "Control Barrier Function-based Quadratic Programs Introduce Undesirable Asymptotically Stable Equilibria". In: *IEEE Control Systems Letters* 5.2 (2020), pp. 731–736.

[164] David Silver et al. "Mastering the Game of Go with Deep Neural Networks and Tree Search". In: *Nature* 529.7587 (2016), pp. 484–489.

[165] David Silver et al. "Mastering the Game of Go without Human Knowledge". In: *Nature* 550.7676 (2017), pp. 354–359.

[166] Julian Schrittwieser et al. "Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model". In: *Nature* 588.7839 (2020), pp. 604–609.

[167] Oriol Vinyals et al. "Alphastar: Mastering the Real-time Strategy Game Starcraft II". In: *DeepMind Blog* 2 (2019), p. 20.

[168] Christopher Berner et al. "Dota 2 with Large Scale Deep Reinforcement Learning". In: *arXiv preprint arXiv:1912.06680* (2019).

[169] Deheng Ye et al. "Towards Playing Full Moba Games with Deep Reinforcement Learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 621–632.

[170] Greg Brockman et al. "Openai Gym". In: *arXiv preprint arXiv:1606.01540* (2016).

[171] Yuval Tassa et al. "Deepmind Control Suite". In: *arXiv preprint arXiv:1801.00690* (2018).

[172] Jemin Hwangbo et al. "Learning Agile and Dynamic Motor Skills for Legged Robots". In: *Science Robotics* 4.26 (2019), eaau5872.

[173] E. Altman. *Constrained Markov Decision Processes*. Stochastic Modeling Series. Taylor & Francis, 1999. ISBN: 9780849303821.

[174] Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[175] Yu Fan Chen et al. "Decentralized Non-Communicating Multiagent Collision Avoidance with Deep Reinforcement Learning". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 285–292.

[176] Dongjie Yu et al. “Reachability Constrained Reinforcement Learning”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 25636–25655.

[177] Alex Ray, Joshua Achiam, and Dario Amodei. *Benchmarking Safe Exploration in Deep Reinforcement Learning*. 2019. arXiv: 1910.01708.

[178] Peter Geibel and Fritz Wysotzki. “Risk-Sensitive Reinforcement Learning Applied to Control under Constraints”. In: *Journal of Artificial Intelligence Research* 24 (2005), pp. 81–108.

[179] Peter Geibel. “Reinforcement Learning for MDPs with Constraints”. In: *17th European Conference on Machine Learning*. Springer. 2006, pp. 646–653.

[180] Weiye Zhao et al. “State-wise Safe Reinforcement Learning: A Survey”. In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*. Survey Track. Aug. 2023, pp. 6814–6822.

[181] Oswin So and Chuchu Fan. “Solving Stabilize-Avoid Optimal Control via Epigraph Form and Deep Reinforcement Learning”. In: *Proceedings of Robotics: Science and Systems*. Daegu, Republic of Korea, 2023.

[182] Yaodong Yang and Jun Wang. “An Overview of Multi-Agent Reinforcement Learning from Game Theoretical Perspective”. In: *arXiv preprint arXiv:2011.00583* (2020).

[183] Yu Fan Chen et al. “Socially Aware Motion Planning with Deep Reinforcement Learning”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 1343–1350.

[184] Samaneh Hosseini Semnani et al. “Multi-Agent Motion Planning for Dense and Dynamic Environments via Deep Reinforcement Learning”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3221–3226.

[185] Ryan Lowe et al. “Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments”. In: *Advances in Neural Information Processing Systems* 30 (2017).

[186] Chenyi Liu et al. “CMIX: Deep Multi-Agent Reinforcement Learning with Peak and Average Constraints”. In: *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference*. Springer. 2021, pp. 157–173.

[187] Dongsheng Ding et al. “Provably Efficient Generalized Lagrangian Policy Optimization for Safe Multi-Agent Reinforcement Learning”. In: *Learning for Dynamics and Control Conference*. PMLR. 2023, pp. 315–332.

[188] Nan Geng et al. “A Reinforcement Learning Framework for Vehicular Network Routing Under Peak and Average Constraints”. In: *IEEE Transactions on Vehicular Technology* (2023).

[189] Yongxin Chen. “Density Control of Interacting Agent Systems”. In: *IEEE Transactions on Automatic Control* (2023).

[190] Zengyi Qin, Yuxiao Chen, and Chuchu Fan. “Density Constrained Reinforcement Learning”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8682–8692.

[191] Pinxin Long et al. “Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 6252–6259.

[192] Pierre-François Massiani et al. “Safe Value Functions”. In: *IEEE Transactions on Automatic Control* 68.5 (2023), pp. 2743–2757.

[193] Guillaume Sartoretti et al. “PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning”. In: *IEEE Robotics and Automation Letters* 4.3 (2019), pp. 2378–2385.

[194] Mehul Damani et al. “PRIMAL _2: Pathfinding via Reinforcement and Imitation Multi-Agent Learning-lifelong”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 2666–2673.

[195] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. “Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving”. In: *arXiv preprint arXiv:1610.03295* (2016).

[196] Vivek S Borkar. “An Actor-Critic Algorithm for Constrained Markov Decision Processes”. In: *Systems & Control Letters* 54.3 (2005), pp. 207–213.

[197] Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. "Reward Constrained Policy Optimization". In: *International Conference on Learning Representations*. 2019.

[198] Joshua Achiam et al. "Constrained Policy Optimization". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 22–31.

[199] Herbert Robbins and Sutton Monro. "A Stochastic Approximation Method". In: *The Annals of Mathematical Statistics* (1951), pp. 400–407.

[200] Vivek S Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Vol. 48. Springer, 2009.

[201] Harsh Satija, Philip Amortila, and Joelle Pineau. "Constrained Markov Decision Processes via Backward Value Functions". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 8502–8511.

[202] John Schulman et al. "Trust Region Policy Optimization". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 1889–1897.

[203] Jakub Grudzien Kuba et al. "Trust Region Policy Optimisation in Multi-Agent Reinforcement Learning". In: *International Conference on Learning Representations*. 2022.

[204] Tabish Rashid et al. "Monotonic value Function Factorisation for Deep Multi-Agent Reinforcement Learning". In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 7234–7284.

[205] Jean-Michel Lasry and Pierre-Louis Lions. "Mean Field Games". In: *Japanese Journal of Mathematics* 2.1 (2007), pp. 229–260.

[206] Alain Bensoussan, Jens Frehse, Phillip Yam, et al. *Mean Field Games and Mean Field type Control Theory*. Vol. 101. Springer, 2013.

[207] Alex Tong Lin et al. "Alternating the Population and Control Neural Networks to Solve High-Dimensional Stochastic Mean-Field Games". In: *Proceedings of the National Academy of Sciences* 118.31 (2021).

[208] Guan-Horng Liu et al. "Deep Generalized Schrödinger bridge". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 9374–9388.

[209] Anthony Corso et al. “A Survey of Algorithms for Black-Box Safety Validation of Cyber-Physical Systems”. In: *Journal of Artificial Intelligence Research* 72 (2021), pp. 377–428.

[210] Rajeev Alur. “Formal Verification of Hybrid Systems”. In: *Proceedings of the Ninth ACM International Conference on Embedded Software*. 2011, pp. 273–278.

[211] Matthias Althoff, Goran Frehse, and Antoine Girard. “Set Propagation Techniques for Reachability Analysis”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 4 (2021), pp. 369–395.

[212] Xin Chen and Sriram Sankaranarayanan. “Reachability Analysis for Cyber-Physical Systems: Are We There Yet?” In: *NASA Formal Methods Symposium*. Springer. 2022, pp. 109–130.

[213] Changliu Liu et al. “Algorithms for verifying deep neural networks”. In: *Foundations and Trends in Optimization* 4.3-4 (2021), pp. 244–404.

[214] Alessandro Abate et al. “FOSSIL: A Software Tool for the Formal Synthesis of Lyapunov Functions and Barrier Certificates Using Neural Networks”. In: *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*. New York, NY, USA: Association for Computing Machinery, 2021.

[215] Hongkai Dai et al. “Lyapunov-stable Neural Network Control”. In: *Proceedings of Robotics: Science and Systems*. 2021.

[216] Guillaume P Brat et al. *Autonomy Verification & Validation Roadmap and Vision 2045*. Tech. rep. NASA/TM-20230003734. 2023.

[217] R. Alur et al. “Discrete Abstractions of Hybrid Systems”. In: *Proceedings of the IEEE* 88.7 (2000), pp. 971–984.

[218] Xinrui Wang, Karen Leung, and Marco Pavone. “Infusing Reachability-Based Safety into Planning and Control for Multi-agent Interactions”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 6252–6259.

[219] Matthew Cavorsi et al. “Multi-Robot Adversarial Resilience using Control Barrier Functions”. In: *Proceedings of Robotics: Science and Systems*. New York City, NY, USA, 2022.

[220] Mo Chen, Jennifer C. Shih, and Claire J. Tomlin. "Multi-vehicle Collision Avoidance via Hamilton-Jacobi Reachability and Mixed Integer Programming". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. 2016, pp. 1695–1700.

[221] Marco Sälzer and Martin Lange. "Fundamental Limits in Formal Verification of Message-Passing Neural Networks". In: *The Eleventh International Conference on Learning Representations*. 2023.

[222] Songyuan Zhang et al. "Compositional Neural Certificates for Networked Dynamical Systems". In: *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*. PMLR, 2023, pp. 272–285.

[223] Shen Shen and Russ Tedrake. "Compositional Verification of Large-Scale Nonlinear Systems via Sums-of-Squares Optimization". In: *2018 Annual American Control Conference (ACC)*. 2018, pp. 4385–4392.

[224] Matthias Althoff. "Formal and Compositional Analysis of Power Systems Using Reachable Sets". In: *IEEE Transactions on Power Systems* 29.5 (2014), pp. 2270–2280.

[225] Dan Zhang et al. "Physical Safety and Cyber Security Analysis of Multi-Agent Systems: A Survey of Recent Advances". In: *IEEE/CAA Journal of Automatica Sinica* 8.2 (2021), pp. 319–333.

[226] Kunal Garg et al. "Model-Free Neural Fault Detection and Isolation for Safe Control". In: *IEEE Control Systems Letters* 7 (2023), pp. 3169–3174.

[227] Guang Wang et al. "Using Machine Learning for Determining Network Robustness of Multi-Agent Systems Under Attacks". In: *PRICAI 2018: Trends in Artificial Intelligence*. Ed. by Xin Geng and Byeong-Ho Kang. Cham: Springer International Publishing, 2018, pp. 491–498.

[228] Arturo Servin and Daniel Kudenko. "Multi-agent Reinforcement Learning for Intrusion Detection". In: *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*. Ed. by Karl Tuyls et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 211–223.

[229] Yoriyuki Yamagata et al. "Falsification of Cyber-Physical Systems Using Deep Reinforcement Learning". In: *IEEE Transactions on Software Engineering* 47.12 (2021), pp. 2823–2840.

[230] Antonis Papachristodoulou, Ali Jadbabaie, and Ulrich Münz. "Effects of Delay in Multi-Agent Consensus and Oscillator Synchronization". In: *IEEE Transactions on Automatic Control* 55.6 (2010), pp. 1471–1477.

[231] Jiechuan Jiang and Zongqing Lu. "Learning Attentional Communication for Multi-Agent Cooperation". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[232] Rundong Wang et al. "Learning Efficient Multi-agent Communication: An Information Bottleneck Approach". In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. PMLR, 2020, pp. 9908–9918.

[233] Abhishek Das et al. "TarMAC: Targeted Multi-Agent Communication". In: *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019, pp. 1538–1546.

[234] Woojun Kim, Jongeui Park, and Youngchul Sung. "Communication in Multi-Agent Reinforcement Learning: Intention Sharing". In: *International Conference on Learning Representations*. 2021.

[235] Devansh R Agrawal and Dimitra Panagou. "Safe Control Synthesis via Input Constrained Control Barrier Functions". In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 6113–6118.

[236] Wenceslao Shaw Cortez, Xiao Tan, and Dimos V Dimarogonas. "A Robust, Multiple Control Barrier Function Framework for Input Constrained Systems". In: *IEEE Control Systems Letters* 6 (2021), pp. 1742–1747.

[237] Geraud Nangue Tasse et al. "ROSARL: Reward-Only Safe Reinforcement Learning". In: *arXiv preprint arXiv:2306.00035* (2023).